

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-027752
 (43)Date of publication of application : 28.01.1997

(51)Int.Cl. H03M 7/30
 H03M 7/40
 H04N 1/41
 H04N 7/24

(21)Application number : 08-172682 (71)Applicant : RICOH CO LTD
 (22)Date of filing : 02.07.1996 (72)Inventor : AAMADO ZANDEI
 EDOWAADO ERU SHIYUWARUTSU
 MAIKERU JIEE GOOMITSUSHIYU
 MAATEIN BOORITSUKU

(30)Priority
 Priority number : 95 498695 Priority date : 03.07.1995 Priority country : US

(54) ENCODER, ENCODING METHOD, DECODING METHOD AND DECODER

(57)Abstract:

PROBLEM TO BE SOLVED: To utilize a reversible wavelet filter for a non-losable/losable data compression system.

SOLUTION: At the encoding part of the non-losable/losable data compression system, a reversible wavelet filter 102 converts input image data to plural coefficients. A ranking/modeling block 103 generates an embedded code stream from the plural coefficients. An entropy encoder 104 generates a compressed data stream by performing entropy encoding to the embedded code stream. The reversible wavelet filter 102 performs the efficient conversion of precise restoration provided by integer arithmetic.



LEGAL STATUS

[Date of request for examination] 03.10.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3367825
[Date of registration] 08.11.2002
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-27752

(43)公開日 平成9年(1997)1月28日

(51) Int.Cl. ⁶	識別記号	序内整理番号	F I	技術表示箇所
H 03 M	7/30	9382-5K	H 03 M	7/30
	7/40	9382-5K		7/40
H 04 N	1/41		H 04 N	1/41
	7/24			7/13

審査請求 未請求 請求項の数67 O.L (全 50 頁)

(21)出願番号 特願平8-172682

(22)出願日 平成8年(1996)7月2日

(31)優先権主張番号 08/498695

(32)優先日 1995年7月3日

(33)優先権主張国 米国(US)

(71)出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72)発明者 アーマド ザンディ

アメリカ合衆国 カリフォルニア州

94025 メンローパーク サンド ヒル

ロード 2882 リコーコーポレーション内

(72)発明者 エドワード エル シュワルツ

アメリカ合衆国 カリフォルニア州

94025 メンローパーク サンド ヒル

ロード 2882 リコーコーポレーション内

(74)代理人 弁理士 鈴木 誠 (外1名)

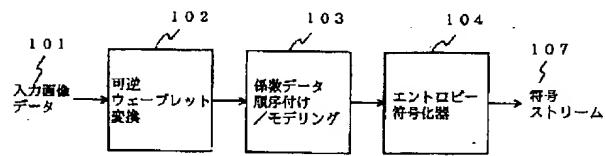
最終頁に続く

(54)【発明の名称】 符号化装置、符号化方法、復号方法及び復号装置

(57)【要約】

【課題】 非損失性／損失性のデータ圧縮システムのための、可逆ウェーブレットフィルタを利用した効率的なデータ符号化／復号方法及び装置を実現する。

【解決手段】 非損失性／損失性圧縮システムの符号化部において、可逆ウェーブレットフィルタ102は入力画像データを複数の係数に変換する。順序付け／モデリングブロック103は複数の係数から埋め込み符号ストリームを生成する。エントロピー符号化器104は埋め込み符号ストリームをエントロピー符号化して圧縮データストリームを生成する。可逆ウェーブレットフィルタ102は整数演算で実現される精密復元の効率的変換である。



【特許請求の範囲】

【請求項1】 入力データを圧縮データストリームに符号化する符号化装置であって、

該入力データを複数の係数へ変換する可逆ウェーブレットフィルタ、

該可逆ウェーブレットフィルタと結合され、該複数の係数に応じて埋め込み符号ストリームを生成する順序付け及びモデリング機構、及び該順序付け及びモデリング機構と結合され、該埋め込み符号ストリームをエントロピー符号化して該圧縮データストリームを発生するエントロピー符号化器を具備する符号化装置。

【請求項2】 請求項1記載の符号化装置において、該可逆ウェーブレットフィルタ、該順序付け及びモデリング機構、及び該2元エントロピー符号化器は因果関係をもって動作し該入力データを1パスで圧縮することを特徴とする符号化装置。

【請求項3】 請求項1記載の符号化装置において、該可逆ウェーブレットフィルタは少なくとも1つのS変換を含むことを特徴とする符号化装置。

【請求項4】 請求項1記載の符号化装置において、該可逆ウェーブレットフィルタは少なくとも1つのTS変換を含むことを特徴とする符号化装置。

【請求項5】 請求項1記載の符号化装置において、該入力データに対し垂直方向及び水平方向に異なった変換が適用されることを特徴とする符号化装置。

【請求項6】 請求項1記載の符号化装置において、該可逆ウェーブレットフィルタは水平パスに対しTS変換を適用し、垂直パスに対し少なくとも1つのS変換と少なくとも1つのTS変換を適用することを特徴とする符号化装置。

【請求項7】 請求項1記載の符号化装置において、該可逆ウェーブレットフィルタは該入力データの行及び列に沿って別々に1次元(1-D)フィルタを適用することにより複数の係数を生成することを特徴とする符号化装置。

【請求項8】 請求項1記載の符号化装置において、該可逆ウェーブレットフィルタはピラミッド分割を実行することを特徴とする符号化装置。

【請求項9】 請求項1記載の符号化装置において、該順序付け及びモデリング機構は該埋め込み符号ストリームを生成するために該複数の係数を順序付けしつつ該複数の係数内の2進値を順序付けすることを特徴とする符号化装置。

【請求項10】 請求項9記載の符号化装置において、該複数係数中の各係数内の2進値は埋め込み順序付けされることを特徴とする符号化装置。

【請求項11】 請求項10記載の符号化装置において、該複数係数中の各係数内の2進値はビット有意度に従って順序付けされることを特徴とする符号化装置。

【請求項12】 請求項9記載の符号化装置において、

該複数係数は、量子化を考慮してビットプレーン符号化の前に相互に整列されることを特徴とする符号化装置。

【請求項13】 請求項12記載の符号化装置において、量子化される重さが少ない係数ほど早いビットプレーン側へ整列されることを特徴とする符号化装置。

【請求項14】 請求項9記載の符号化装置において、符号ビットは最初の非ゼロの絶対値ビットと一緒に符号化されることを特徴とする符号化装置。

【請求項15】 請求項1記載の符号化装置において、該順序付け及びモデリング機構は符号化単位内のビットを、係数の空間及びスペクトル従属性に基づきモデリングし、該2元エントロピー符号化器は該順序付け及びモデリング機構によりモデリングされたビットに基づいて符号化を行うことを特徴とする符号化装置。

【請求項16】 請求項15記載の符号化装置において、該順序付け及びモデリング機構は隣接係数及び親係数に基づくコンテキストを使ってビットをモデリングすることを特徴とする符号化装置。

【請求項17】 請求項16記載の符号化装置において、該コンテキストは因果関係的であることを特徴とする符号化装置。

【請求項18】 請求項1記載の符号化装置において、該エントロピー符号化器は2元エントロピー符号化器からなることを特徴とする符号化装置。

【請求項19】 請求項18記載の符号化装置において、該エントロピー符号化器はQ-コーダーからなることを特徴とする符号化装置。

【請求項20】 請求項18記載の符号化装置において、該エントロピー符号化器はQM-コーダーからなることを特徴とする符号化装置。

【請求項21】 請求項18記載の符号化装置において、該エントロピー符号化器は有限状態マシンコーダーからなることを特徴とする符号化装置。

【請求項22】 請求項1記載の符号化装置において、該エントロピー符号化器は並列コーダーからなることを特徴とする符号化装置。

【請求項23】 入力データを圧縮データストリームに符号化する符号化装置であって、該入力データを複数の係数に変換する可逆ウェーブレットフィルタ、

該可逆ウェーブレットフィルタと結合され、該複数の係数に対する埋め込み量子化を行って、該複数の係数に応じて埋め込み符号ストリームを生成する順序付け及びモデリング機構、及び該順序付け及びモデリング機構と結合され、該埋め込み符号ストリームを2元エントロピー符号化して該圧縮データストリームを生成する2元エントロピー符号化器、を具備する符号化装置。

【請求項24】 請求項23記載の符号化装置において、該可逆ウェーブレットフィルタは水平変換と垂直変換を交互に適用して複数レベルの分離可能なピラミッド

分割を実行し、かつ、該可逆ウェーブレットフィルタは TS 変換だけにより水平分割を実行し、また TS 変換と S 変換の組合せにより垂直変換を実行することを特徴とする符号化装置。

【請求項 25】 請求項 24 記載の符号化装置において、該可逆ウェーブレットフィルタは 2 つの TS 変換とそれに続く 2 つの S 変換により垂直変換を実行することを特徴とする符号化装置。

【請求項 26】 請求項 23 記載の符号化装置において、該順序付け及びモデリング機構はヘッダの後に上から下へ順に続いた符号化単位を用いて符号ストリームを順序付けすることを特徴とする符号化装置。

【請求項 27】 請求項 26 記載の符号化装置において、各符号化単位は、符号化されないラスター順の LL 係数と、それに続く 1 ビットプレーンずつエンタロピー符号化されたデータとからなることを特徴とする符号化装置。

【請求項 28】 請求項 27 記載の符号化装置において、各符号化ユニット内のエンタロピー符号化データは、最上位ビットプレーンから最下位ビットプレーンへと順序付けられることを特徴とする符号化装置。

【請求項 29】 請求項 23 記載の符号化装置において、該順序付け及びモデリング機構は係数アラインメントを実行することを特徴とする符号化装置。

【請求項 30】 請求項 29 記載の符号化装置において、各サブロックの係数は各符号化単位のヘッダ中の指示に従って整列されることを特徴とする符号化装置。

【請求項 31】 請求項 29 記載の符号化装置において、係数は平均 2 乗誤差の点から整列されることを特徴とする符号化装置。

【請求項 32】 請求項 23 記載の符号化装置において、該順序付け及びモデリング機構において係数はビット有意度別に符号化されることを特徴とする符号化装置。

【請求項 33】 請求項 32 記載の符号化装置において、係数はビット有意度別に符号化され、ここにおいて第 1 ビットプレーンは該係数全部の最も左の絶対値ビットであることを特徴とする符号化装置。

【請求項 34】 請求項 33 記載の符号化装置において、各係数の符号ビットは、当該係数が非ゼロの絶対値を持つ最も高いビットプレーンまで、符号化されないことを特徴とする符号化装置。

【請求項 35】 請求項 23 記載の符号化装置において、該順序付け及びモデリング機構は、各ビットプレーン内の係数を、低解像度・低周波数レベル (LL) から高解像度・高周波数 (HH) レベルへと順序付けることを特徴とする符号化装置。

【請求項 36】 請求項 35 記載の符号化装置において、該順序付け及びモデリング機構は係数を、4-L, L, 4-HL, 4-LH, 4-HH, 3-HL, 3-L

H, 3-HH, 2-HL, 2-LH, 2-HH, 1-HL, 1-LH, 1-HH と順序付けることを特徴とする符号化装置。

【請求項 37】 請求項 36 記載の符号化装置において、該順序付け及びモデリング機構は各サブロック内においてラスター順に符号化することを特徴とする符号化装置。

【請求項 38】 請求項 23 記載の符号化装置において、該順序付け及びモデリング機構は隣接係数値及び親係数値に基づくコンテキストを利用するコンテキストモデルを含むことを特徴とする符号化装置。

【請求項 39】 請求項 38 記載の符号化装置において、該コンテキストは、最初の非ゼロ絶対値ビットが見つかったか否かと、該最初の非ゼロ絶対値ビットが見つかった場合の該最初の非ゼロ絶対値ビットからのビットプレーン数に基づいていることを特徴とする符号化装置。

【請求項 40】 請求項 39 記載の符号化装置において、該最初の非ゼロ絶対値ビットが見つかっていない場合、該コンテキストモデルは係数を条件付けするために、当該係数の親に対応した情報及び当該係数の NW, N, NE, W, E, SW, S 又は SE 位置にある少なくとも 1 つの係数からの情報を含むコンテキストを用いることを特徴とする符号化装置。

【請求項 41】 請求項 40 記載の符号化装置において、該コンテキストモデルは係数を条件付けするために、当該係数から 1 レベル以上離れた当該係数の親を含むコンテキストを用いることを特徴とする符号化装置。

【請求項 42】 請求項 40 記載の符号化装置において、該最初の非ゼロ絶対値ビットが見つかっている場合、該コンテキストモデルは

(a) 該非ゼロ絶対値ビットは直前ビットプレーンに見つかった、

(b) 該非ゼロ絶対値ビットは 2 又は 3 ビットプレーン前に見つかった、又は

(c) 該非ゼロ絶対値ビットは 4 ビットプレーン以前に見つかった、を含むコンテキストを用いることを特徴とする符号化装置。

【請求項 43】 請求項 39 記載の符号化装置において、注目係数の符号ビットは該注目係数の注目ビットが該最初の非ゼロ絶対値ビットならばその直後に符号化されることを特徴とする符号化装置。

【請求項 44】 請求項 39 記載の符号化装置において、真北の係数の符号が未知の時に符号ビットを符号化するためのコンテキストは固定確率であることを特徴とする符号化装置。

【請求項 45】 請求項 44 記載の符号化装置において、該固定確率は約 0.5 であることを特徴とする符号化装置。

【請求項 46】 請求項 39 記載の符号化装置において

て、該最初の非ゼロ絶対値ビットの後の2進値を符号化するためのコンテキストは固定確率であることを特徴とする符号化装置。

【請求項47】 請求項46記載の符号化装置において、該最初の非ゼロ絶対値ビットの後の最初の2進値を符号化するためのコンテキストは約0.7の固定確率であることを特徴とする符号化装置。

【請求項48】 請求項39記載の符号化装置において、該最初の非ゼロ絶対値ビットの後の2番目及び3番目の2進値を符号化するためのコンテキストは約0.6の固定確率であることを特徴とする符号化装置。

【請求項49】 請求項39記載の符号化装置において、該最初の非ゼロ絶対値ビットの後の4番目以降の2進値を符号化するためのコンテキストは固定確率0.5で符号化されることを特徴とする符号化装置。

【請求項50】 請求項23記載の符号化装置において、該エントロピー符号化器はQ-コーダーからなることを特徴とする符号化装置。

【請求項51】 請求項23記載の符号化装置において、該エントロピー符号化器はQM-コーダーからなることを特徴とする符号化装置。

【請求項52】 請求項23記載の符号化装置において、該エントロピー符号化器は有限状態マシンコーダーからなることを特徴とする符号化装置。

【請求項53】 請求項23記載の符号化装置において、該エントロピー符号化器は並列コーダーからなることを特徴とする符号化装置。

【請求項54】 入力データを圧縮データストリームへ符号化する方法であって、可逆ウエーブレットフィルタを使って該入力データを複数の係数に変換するステップ、該複数の係数に応じて埋め込み符号ストリームを生成するステップ、及び該埋め込み符号ストリームをエントロピー符号化して該圧縮データを生成するステップ、を有する符号化方法。

【請求項55】 請求項54記載の符号化方法において、該変換のステップ、該生成のステップ及び該エントロピー符号化のステップは因果関係的に実行されることを特徴とする符号化方法。

【請求項56】 請求項54記載の符号化方法において、該変換のステップは該入力データに対し垂直方向及び水平方向に異なった変換を適用することからなることを特徴とする符号化方法。

【請求項57】 該入力データに対し少なくとも1つのTS変換を適用するステップをさらに有することを特徴とする請求項56記載の符号化方法。

【請求項58】 該入力データに対し少なくとも1つのS変換を適用するステップをさらに有することを特徴とする請求項56記載の符号化方法。

【請求項59】 該入力データに対し複数レベルのピラ

ミッド分割を実行するステップであって、水平分割のためにTS変換だけを使い、垂直分割のためにTS変換とS変換の組合せを使って、水平変換と垂直変換を交互に適用するステップを含むステップをさらに有することを特徴とする請求項54記載の符号化方法。

【請求項60】 請求項54記載の符号化方法において、該埋め込み符号ストリームを生成するステップは、該埋め込み符号ストリームを生成するために、該複数の係数を順序付けするステップ及び該複数の係数内の2進値を順序付けするステップを含むことを特徴とする符号化方法。

【請求項61】 符号化データストリームを復号する方法であって、

1符号化単位の符号化データを受け取るステップ、コンテキストモデル及びエントロピー符号化器によって各係数のビットをモデリングするステップ、及び最も粗いレベルから、係数にインバース・ウエーブレットフィルタを適用するステップ、を有する復号方法。

【請求項62】 全レベルがインバータ・フィルタ処理されたか判定し、否ならば次に粗いレベルから係数にインバース・ウエーブレットフィルタを適用するステップをさらに有することを特徴とする請求項61記載の復号方法。

【請求項63】 全レベルがインバース・フィルタ処理されるまで該判定のステップを繰り返すステップをさらに有する請求項62記載の復号方法。

【請求項64】 符号化データを復号する復号装置であって、

該符号化データをエントロピー復号して係数の符号ストリームを生成するエントロピー復号器、及び該エントロピー復号器と結合され、該係数の符号ストリームを復元データに変換するリバース・ウエーブレットフィルタ、を具備する復号装置。

【請求項65】 請求項64記載の復号装置において、該エントロピー復号器は該符号化データを係数の埋め込み符号ストリームに復号することを特徴とする復号装置。

【請求項66】 請求項39記載の符号化装置において、該最初の非ゼロ絶対値ビットが見つかっていないときには、該コンテキストモデルは係数を条件付けするために当該係数の親に対応する情報を含むコンテキストを用いることを特徴とする符号化装置。

【請求項67】 請求項39記載の符号化装置において、該最初の非ゼロ絶対値ビットが見つかっていないときには、該コンテキストモデルは係数を条件付けするために当該係数のNW, N, NE, W, E, SW, S又はSE位置にある少なくとも1つの係数からの情報を含むコンテキストを用いることを特徴とする符号化装置。

【発明の詳細な説明】

【0001】

【発明の属する分野】本発明は、データ圧縮及び伸長システムの分野に係り、特に、圧縮／伸長システムにおけるデータの非損失性 (lossless) 及び損失性 (lossy) の符号化及び復号の方法及び装置に関する。

【0002】

【従来の技術】データ圧縮は、大量のデータの蓄積及び伝送のために非常に有用なツールである。例えば、文書のファクシミリ伝送のような画像伝送に要する時間は、圧縮を利用して画像再生に必要とされるビット数を減らすと飛躍的に短縮される。

【0003】従来より、多くの様々なデータ圧縮手法が存在している。圧縮手法は、おおまかに分類すると 2 つのカテゴリー、つまり損失性符号化と非損失性符号化とに分けることができる。損失性符号化とは、情報の損失を生じ、したがってオリジナルデータの完全な再現が保証されない符号化のことである。損失性符号化の目標とするところは、オリジナルデータから変化しても、その変化が不快であったり目だったりしないようにすることである。非損失性圧縮では、全ての情報が保存され、データは完全な復元が可能な方法で圧縮される。

【0004】非損失性圧縮では、入力シンボルもしくは輝度データが出力符号語に変換される。入力としては、画像データ、音声データ、1 次元データ（例えば空間的または時間的に変化するデータ）、2 次元データ（例えば 2 つの空間軸方向に変化する（または 1 つの空間次元と 1 つの時間次元で変化する）データ）、あるいは多次元／マルチスペクトルのデータがあろう。圧縮がうまくいけば、その符号語は、符号化前の入力シンボル（または輝度データ）のために必要とされるビット数より少ないビット数で表現される。非損失性符号化法には、辞書符号化方式（例えば、Lempel-Ziv 方式）、ランレンジス符号化方式、計数符号化方式、エントロピー符号化方式がある。非損失性の画像圧縮では、圧縮は予測またはコンテキストと符号化に基づいている。ファクシミリ圧縮用 JBIG 規格と、連続階調画像用の DPCM（差分パルス符号変調 - JPEG 規格のオプション）は画像用の非損失性圧縮の例である。損失性圧縮では、入力シンボルまたは輝度データは、量子化されてから出力符号語へ変換される。量子化は、データの重要な特徴量を保存する一方、重要でない特徴量を除去することを目的としている。損失性圧縮システムは、量子化に先立ち、エネルギー集中をするための変換を利用することが多い。JPEG は画像データ用の損失性符号化法の一例である。

【0005】画像信号処理における最近の開発は、効率的かつ高精度のデータ圧縮符号化方式を追求することに関心を集中し続けている。変換またはピラミッド信号処理の様々な方式が提案されており、その中にマルチ解像度ピラミッド処理方式とウェーブレット (wavelet) ピラミッド処理方式がある。これら 2 方式は、サブバンド処理方式及び階層処理方式とも呼ばれる。画像データ

のウェーブレット・ピラミッド処理方式は、直交ミラーフィルタ (QMF) を用いてオリジナル画像のサブバンド分割をする特殊なマルチ解像度ピラミッド処理方式である。なお、他の非QMF ウェーブレット方式もある。ウェーブレット処理方式に関し、これ以上的情報を得るには、Antonini, M., et al., "Image Coding Using Wavelet Transform", IEEE Transactions on Image Processing, Vol. 1, No. 2, April 1992、及び、Shapiro, J., "An Embedded Hierarchical Image Coder Using Zerotrees of Wavelet Coefficients", Proc. IEEE Data Compression Conference, pg s. 214-223, 1993 を参照されたい。

【0006】

【発明が解決しようとする課題】従来の多くのウェーブレット処理方式の問題点は、データ全部を処理中に記憶しておくために大きなメモリが必要となることである。すなわち、ウェーブレット処理を実行する場合、データ全部を、その符号化がなされる前に調べなければならない。その場合、データ全部について少なくとも 1 フルパスを完了するまで全くデータ出力がない。実際には、ウェーブレット処理はデータの複数のパスを必要とするのが普通である。そのため、しばしば大きなメモリが必要になる。大きなメモリを必要とせずに、ウェーブレット処理を利用できることが望ましい。さらに、データの 1 パスだけでウェーブレット処理を実行できるとよい。

【0007】ウェーブレットまたはサブバンド変換装置の多くは、特定の正規型のフィルタを必要とする。例えば、低域通過フィルタと高域通過フィルタは同一長でなければならず、その係数の 2 乗和は 1 でなければならず、高域通過フィルタは時間・周波数特性が低域通過フィルタと正反対でなければならない、等々である。（1991 年 5 月に Lawton 等に発行された米国特許第 5,014,134 号を参照）。より広範な種類のフィルタを許容することが望ましい。すなわち、低域通過フィルタと高域通過フィルタが同一長でなく、その係数の 2 乗和は 1 である必要がなく、高域通過フィルタが低域通過フィルタと正反対の時間・周波数特性である必要がない等々、そのような低域通過フィルタと高域通過フィルタを使用するウェーブレットまたはサブバンド変換装置を提供することが望ましい。

【0008】本発明は、上述の諸点に鑑みてなされたもので、その主たる目的は、非損失性／損失性の圧縮／伸長システムのための可逆ウェーブレットフィルタを利用した効率的な符号化／復号方法及び装置を提供することである。これ以外の本発明の目的は、以下の説明から明らかになろう。

【0009】

【課題を解決するための手段】本発明によれば、入力データを圧縮データストリームに符号化する符号化装置は、該入力データを複数の係数へ変換する可逆ウェーブ

レットフィルタと、該可逆ウエーブレットフィルタと結合され、該複数の係数に応じて埋め込み符号ストリームを生成する順序付け及びモデリング機構と、該順序付け及びモデリング機構と結合され、該埋め込み符号ストリームをエントロピー符号化して該圧縮データストリームを発生するエントロピー符号化器とを具備する構成とされる。

【0010】本発明によれば、入力データを圧縮データストリームに符号化する符号化装置は、該入力データを複数の係数に変換する可逆ウエーブレットフィルタと、該可逆ウエーブレットフィルタと結合され、該複数の係数に対する埋め込み量子化を行って、該複数の係数に応じて埋め込み符号ストリームを生成する順序付け及びモデリング機構と、該順序付け及びモデリング機構と結合され、該埋め込み符号ストリームを2元エントロピー符号化して圧縮データストリームを生成する2元エントロピー符号化器とを具備する構成とされる。

【0011】本発明によれば、入力データを圧縮データストリームへ符号化する符号化方法は、可逆ウエーブレットフィルタを使って該入力データを複数の係数に変換するステップと、該複数の係数に応じて埋め込み符号ストリームを生成するステップと、該埋め込み符号ストリームをエントロピー符号化して該圧縮データを生成するステップとを有する構成とされる。

【0012】本発明によれば、符号化データストリームを復号する復号方法は、1符号化単位の符号化データを受け取るステップと、コンテキストモデル及びエントロピー符号化器によって各係数のビットをモデリングするステップと、最も粗いレベルから、係数にインバース・ウエーブレットフィルタを適用するステップとを有する構成とされる。

【0013】本発明によれば、符号化データを復号する復号装置は、該符号化データをエントロピー復号して係数の符号ストリームを生成するエントロピー復号器と、該エントロピー復号器と結合され、該係数の符号ストリームを復元データに変換するリバース・ウエーブレットフィルタとを具備する構成とされる。

【0014】上記及びその他の本発明の態様並びに特徴については、以下の実施例に関連した本発明の詳細な説明により明白になろう。

【0015】

【発明の実施の形態】以下に、圧縮及び伸長システムについて述べる。圧縮システムにおいて、符号化装置が入力データを圧縮データストリームに符号化する。一実施例では、この符号化装置は、可逆ウエーブレットフィルタ、順序付け／モデリング機構、エントロピー符号化器からなる。可逆ウエーブレットフィルタは、入力データを複数の係数に変換する。順序付け／モデリング機構は、それら係数を受け取り、埋め込み符号ストリームを生成する。エントロピー符号化器は、その埋め込み符号

ストリームをエントロピー符号化して圧縮データストリームを生成する。

【0016】以下の本発明に関する詳細な説明において、本発明を完全に理解してもらうために、符号化器の種類、ビット数、信号名等々、様々な具体例が示される。しかし、当業者には、そのような具体例によらずに本発明を実施し得ることは明白になろう。他方、本発明をいたずらに難解にしないため、周知の構造及びデバイスはブロック図として示し、詳しくは示さない。

【0017】以下の詳細説明のかなりの部分は、コンピュータメモリ内のデータビットに対する演算のアルゴリズム及び記号表現によって与えられる。このようなアルゴリズム記述及び表現は、データ処理技術分野の当業者によって、その研究の内容を他の当業者に対し最も効率的に伝えるために用いられる手段である。あるアルゴリズムがあり、それが概して、希望する結果に至る自己矛盾のないステップ系列だと考えられるでしょう。これらのステップは、物理量の物理的処理を必要とするものである。必ずという訳ではないが、通常、これらの物理量は記憶、転送、結合、比較、その他処理が可能な電気的または磁気的信号の形をとる。これらの信号をビット、値、要素、記号、文字、用語、数字等で表わすのが、主に慣用上の理由から、時に都合がよいことが分かっている。

【0018】しかしながら、このような用語は、適切な物理量と関係付けられるべきであって、これら物理量につけた便宜上のラベルに過ぎないということに留意すべきである。以下の説明から明らかのように、特に断わらない限り、"処理" "演算" "計算" "判定" "表示" 等々の用語を用いて論じることは、コンピュータシステムのレジスタ及びメモリ内の物理的（電子的）な量として表現されたデータを処理して、コンピュータシステムのメモリまたはレジスタ、同様の情報記憶装置、情報伝送装置あるいは表示装置の内部の同様に物理量として表現された他のデータへ変換する、コンピュータシステムあるいは同様の電子演算装置の作用及びプロセスを指すものである。

【0019】本発明はまた、本明細書に述べる操作を実行するための装置にも関係する。この装置は、必要な目的のために専用に作られてもよいし、あるいは、汎用コンピュータを内蔵プログラムにより選択的に駆動または再構成したものでもよい。本明細書に提示されるアルゴリズム及び表示は、本質的に、いかなる特定のコンピュータやその他装置とも関係がない。様々な汎用マシンを本明細書に述べたところに従ったプログラムで利用してもよいし、あるいは、必要な方法のステップの実行のためにより特化した装置を作るほうが好都合であるかもしれない。これら多様なマシンに要求される構造は以下の説明より明らかになろう。さらに、本発明を説明するにあたり、どのような格別のプログラミング言語とも関連

付けない。本明細書において述べるように、本発明の教えるところを実現するために多様なプログラミング言語を使用してよいことが理解されよう。

【0020】下記用語が以下の説明に用いられる。それら各種用語には、すでに語義がある。しかし、規定された語義は、それら用語が当該分野において知られている範囲に限定して考えられるべきでない。これら語義は、本発明の理解を助けるために規定されたものである。

【0021】ビット有意度 (bit-significance) : 符号 (sign) / 絶対値表現に似た数表現で、ヘッド (head) ビットの後に符号 (sign) ビットが続き、さらに、テール (tail) ビットがあれば、その後に続く。埋め込み (embedding) は、この数表現に関しビットプレーン順に符号化する。

【0022】符号化単位 : 一括して符号化される係数の単位で、任意の順序にことができる。一実施例では、符号化単位は矩形に配列された1つ以上のツリーからなる。符号化単位は一つの画像全体、複数の画像の集合、あるいは他のデータ集合から構成されるかもしれない。符号化単位は、変換を実行するために必要なバッファサイズに大きな影響を与える。また、一実施例では、現在符号化単位以外の係数からは全く内容を引き出すことができない。しかし、エントロピー符号は、一つの符号化単位内で、あるいは多数の符号化単位の後で、リセットされても構わない。符号化単位は必ずしもランダムアクセス可能ではない。

【0023】コンテキストモデル : 符号化しようとする現在ビットに関する得られる情報で、現在ビットに関する過去に学習した情報を提供する。これによりエントロピー符号化のための条件付確率予測が可能になる。

【0024】ツリー : 最高レベルのウエーブレット分割のLL内の単一の係数と関連した複数の係数及び画素。係数の個数はレベル数と同じである。

【0025】バンド : 2次元データに対する最高レベルのウエーブレット分割のLL内の1行もしくは1ラインの係数に関する係数及び画素。他次元のデータに対しても、バンドは同様に定義される。

【0026】分割レベル : ウエーブレット分割ピラミッドにおける位置。

【0027】埋め込み量子化 : 符号ストリームに必然的に含まれる量子化。例えば、有意度レベルが、最高有意度レベルから最低有意度レベルへ順に並べられているときには、符号ストリームの単なる切り捨てによって量子化が行われる。タグ、マーカー、ポインタ等によって同じ作用を得ることができる。

【0028】エントロピー符号化器 : 現在ビットを、そのコンテキストに基づいて符号化する装置。コンテキストは、現在ビット (1ビットまたは複数ビット) の最適表現のための確率予測を与える。

【0029】固定レート : ある一定の画素レートを維持

し、バンド幅の限られた通信路を持つアプリケーション又はシステム。これは、全体的に平均して圧縮するよりも、局的に平均して圧縮することを必要とする。例 : MPEG 固定サイズ : 限られたサイズのバッファを持つアプリケーション又はシステム。このような場合、全体的に平均した圧縮が達成される、例えば、プリントバッファ (アプリケーションは、固定レートでかつ固定サイズのことと、そのどちらかのこともある)。

【0030】固定長 : 特定のデータブロックを特定の圧縮データブロックへ変換するシステム (例えば BT C)。固定長符号は固定レート・固定サイズのアプリケーションに適するが、レート-歪み性能は可変レートシステムに比べると劣ることが多い。

【0031】水平コンテキストモデル : エントロピー符号化器用のコンテキストモデルであり (一実施例)、本明細書中で本発明の一部として定義される。

【0032】ヘッド : ビット有意度表現において、ヘッドビットとは最上位ビットから最初の非ゼロのビットまでの絶対値ビットである。

【0033】オーバーラップ変換 : 単一のソース標本点が同一周波数の複数の係数に寄与する変換。その例に、多くのウエーブレットとオーバーラップ直交変換がある。

【0034】プログレッシブ : データの一部から矛盾のない結果を得られ、かつデータを増やすことで精度を上げることができるように順序付けられた符号ストリーム。実施例によっては、データのビットプレーンの浅いほうから深いほうへ順序付けられた符号ストリーム、この場合は普通、ウエーブレット係数データをさす。

【0035】ピラミッド : 解像度が下がる毎に大きさが2分の1 (面積では4分の1) になる一連の解像度成分。

【0036】可逆変換 : 精密な再現性を有する、整数演算により実行される効率的な変換。

【0037】S変換 : 1つの2タップ低域通過フィルタと1つの2タップ高域通過フィルタによる特殊な可逆ウエーブレットフィルタ対。

【0038】テール : ビット有意度表現で、テール (tail) ビットとは最上位の非ゼロのビットより有意度が低い絶対値ビットである。

【0039】テール情報 : 一実施例では、ビット有意度表現で表された係数のためにとり得る4つの状態。係数及び現在ビットプレーンの関数であり、水平コンテキストモデルのために利用される。

【0040】テール・オン (tail-on) : 一実施例では、テール情報状態がゼロか非ゼロであるかに依存する2つの状態。水平コンテキストモデルのために利用される。

【0041】TS変換 : Two-Six変換。1つの2タップ低域通過フィルタと1つの6タップ高域通過フィルタによる特殊なウエーブレットフィルタ対。

【0042】統合型非損失性／損失性：同じ圧縮システムが、非損失性又は損失性の復元が可能な符号化データストリームを提供する。後述の本発明のケースでは、このような符号ストリームは、符号化器に対し設定したり命令したりすることなく、非損失性復元と損失性復元の両方が可能である。

【0043】視覚的有意度レベル：特定のシステムを定義することにより、入力データ（画素データ、係数、誤差信号等）は視覚的効果が同じ複数のグループに論理的に分類される。例えば、最上位の一つまたは複数のビットプレーンは、恐らく下位のビットプレーンより視覚的に重要であろう。また、低い周波数の情報は一般に高い周波数の情報より重要である。”視覚的有意度”的実用的な定義の殆どは、後述のように本発明も含め、何らかの誤差距離に関する。しかし、それよりも良好な視覚的尺度が、視覚的有意度のシステム定義に組み入れられるかもしれない。データの種類が異なれば視覚的有意度レベルも違う。例えば、音声データは音声の有意度レベルを持つ。

【0044】ウェーブレットフィルタ：ウェーブレット変換に使われる高域通過と低域通過の合成フィルタ及び分析フィルタ。

【0045】ウェーブレット変換：“周波数”及び“時間（空間）”領域の両方の拘束条件を用いる変換。

【0046】説明する一実施例では、1つの高域通過フィルタと1つの低域通過フィルタからなる変換である。結果として得られる係数は2:1の間引きを施され（臨界フィルタ処理され）、そして、それらフィルタが低域通過係数に適用される。

【0047】<概要>本発明は、符号化部及び復号部を持つ圧縮／伸長システムを提供する。符号化部は入力データを符号化して圧縮データを生成する役割を持ち、他方、復号部は既に符号化されたデータを復号してオリジナル入力データの復元データを生成する役割を持つ。入力データには、画像（静止画像あるいはビデオ画像）、音声等々の様々な種類のデータが含まれる。一実施例ではデータはデジタル信号データであるが、ディジタル化したアナログデータ、テキストデータ形式、その他の形式も可能である。そのデータのソースは、例えば符号化部及び／または復号部のためのメモリまたは通信路である。

【0048】本発明において、符号化部及び／または復号部の構成要素は、ハードウエア又はコンピュータシステム上で利用されるソフトウエアによって実現し得る。本発明は、非損失性の圧縮／伸長システムを提供する。本発明はまた、損失性の圧縮／伸長を実行するようにも構成し得る。

【0049】<本発明システムの概要>図1は、システムの符号化部の一実施例のブロック図である。なお、システムの復号部は逆の順序で動作し、データフローも同

様である。図1において、入力画像データ101がウェーブレット変換ブロック102に受け取られる。ウェーブレット変換ブロック102の出力は係数データ順序付け／モデリングブロック103に接続される。順序付け／モデリングブロック103は、ウェーブレット変換ブロック102の出力に応じて少なくとも1つのビットストリームを出力し、このビットストリームはエントロピー符号化器104に受け取られる。エントロピー符号化器104は、順序付け／モデリングブロック103からの入力に応じて符号ストリーム107を出力する。

【0050】一実施例において、順序付け／モデリングブロック103は、図2に示されるように、符号(sig n)／絶対値フォーマッティングユニット201と、統合空間／周波数(joint space/frequency; JSF)コンテキストモデル202からなる。一実施例では、統合空間／周波数コンテキストモデル202は後述の水平(horizon)コンテキストモデルである。符号／絶対値フォーマッティングユニット201の入力は、ウェーブレット変換符号化ブロック102の出力と接続される。符号／絶対値フォーマッティングユニット202の出力は、統合空間／周波数モデリングブロック202に接続される。JSFコンテキストモデル202の出力はエントロピー符号化器104の入力に接続され、エントロピー符号化器104は出力符号ストリーム107を発生する。

【0051】図1に戻る。本発明においては、後に定義するように、画像データ101はウェーブレット変換ブロック102において受け取られ、可逆ウェーブレットを使って変換符号化され、その画像のマルチ解像度分割を表わす一連の係数が生成される。本発明の可逆ウェーブレット変換は、計算が複雑でなく、ソフトウエア又はハードウエアにより、全く系統誤差を生じさせないで実行できる。さらに、本発明のウェーブレットはエネルギー集中及び圧縮性能に優れている。これらの係数は順序付け／モデリングブロック103に受け取られる。

【0052】順序付け／モデリングブロック103は係数の順序付けとモデリングをする。この係数順序付けにより埋め込み(embedded)データストリームが得られる。この埋め込みデータストリームは、符号化時、伝送時又は復号時に、結果の符号ストリームが量子化されることを許す。一実施例において、順序付け／モデリングブロック103は、それらの係数を、その有意度（後述）に基づき順序付けして符号／絶対値形式へ変換し、このフォーマットされた係数は、埋め込み(embedded)モデリング法を適用される。一実施例では、フォーマットされた係数は統合空間／周波数モデリングを適用される。

【0053】順序付け及びモデリングの結果が、エントロピー符号化器により符号化すべきデシジョン(decisions)（又はシンボル）である。一実施例では、全てのデシジョンが一つの符号化器へ送られる。他の実施例で

は、デシジョンは有意度によってラベル付けされ、デシジョンは有意度レベル毎に別々の（物理的または仮想的な）複数の符号化器によって処理される。

【0054】図2に戻って、JSFコンテキストモデルブロック201から出力されるビットストリームは、エントロピー符号化器104により有意度順に符号化される。一実施例では、エントロピー符号化器104は1つ又は複数の2次元エントロピー符号化器からなる。

【0055】<ウエーブレット分割>本発明は、最初に、可逆ウエーブレットにより、画像（画像データとしての）または他のデータ信号の分割を行なう。本発明において、可逆ウエーブレット変換は、整数係数を持つ信号の非損失性復元が可能な精密復元システムを整数演算で実現する。本発明は、可逆ウエーブレットを利用するにより、有限精度の演算で非損失性圧縮をすることができる。画像データに可逆ウエーブレット変換を適用することにより生成される結果は、一連の係数である。

【0056】本発明の可逆ウエーブレット変換は、フィルタの集合を用いて実現し得る。一実施例では、そのフィルタは1つの2タップ低域通過フィルタと1つの6タップ高域通過フィルタである。一実施例では、これらフィルタは加減算（とハードワイヤのビットシフト）だけで実現される。また、本発明の一実施例では、その高域通過フィルタは低域通過フィルタの結果を使って出力を生成する。結果として得られる高域通過係数は画素深度(depth)より数ビットだけ大きく、低域通過係数は画素深度と同一である。ピラミッド分割では低域通過係数だけが繰り返しフィルタされるため、マルチレベル分割で係数解像度は増加しない。

【0057】他の実施例では、高域通過フィルタの出力係数でなく、低域通過フィルタの出力係数のサイズが増加するかもしれない。

【0058】ウエーブレット変換システムは1対のFIR分析フィルタ $h_0(n)$, $h_1(n)$ と1対のFIR合成フィルタ $g_0(n)$, $g_1(n)$ によって定義される。本発明において、 h_0 と g_0 は低域通過フィルタであり、 h_1 と g_1 は高域通過フィルタである。このウエーブレット変換システムのブロック図が図3に示されている。

【0059】図3において、入力信号 $x(n)$ に対してブロック211, 212で分析フィルタ h_0 , h_1 がかけられ、その出力がブロック213, 214で2:1の間引き（臨界的サブサンプリング）を施されることにより、変換信号 $y_0(n)$, $y_1(n)$ が生成される。この変換信号 $y_0(n)$, $y_1(n)$ はそれぞれ、本明細書においては低域通過（平滑）係数、高域通過（細密）係数と呼ばれる。これらの分析フィルタのブロック211, 212と、それに対応した間引きもしくはサブサンプリングのブロック213, 214は、ウエーブレット変換システムの分析部を構成する。符号化器/復号器215, 216は、変換ドメインにおいて実行される全ての処理ロジック及びル

ーチン（例えば、予測、量子化、符号化等々）を含む。

【0060】図3に示したウエーブレット変換システムは合成部も持ち、この合成部において変換信号はブロック217, 218で1:2のアップサンプリングを施され（各項の後にゼロが挿入される）、ついでブロック219, 220で合成フィルタ $g_0(n)$, $g_1(n)$ に通される。低域通過（平滑）係数 $y_0(n)$ は低域通過合成フィルタ g_0 に通され、高域通過（細密）係数 $y_1(n)$ は高域通過合成フィルタ g_1 に通される。フィルタ $g_0(n)$, $g_1(n)$ の出力が合成され

【0061】

【外1】

$\hat{X}(n)$

【0062】が作られる。

【0063】ある実施例ではダウンサンプリングとアップサンプリングが行なわれるが、他の実施例では、ダウンサンプリング及びアップサンプリングにより不要となる計算が行なわれないようなフィルタが使用される。

【0064】ウエーブレット変換システムはZ変換によって記述してもよい。ここで、

【0065】

【外2】

$X(Z)$, $\hat{X}(Z)$

【0066】はそれぞれ入力信号と出力信号であり、 $Y_0(Z)$, $Y_1(Z)$ は低域通過の変換信号と高域通過の変換信号であり、 $H_0(Z)$, $H_1(Z)$ は低域通過分析フィルタと高域通過分析フィルタであり、 $G_0(Z)$, $G_1(Z)$ は低域通過合成フィルタと高域通過合成フィルタである。変換ドメインに修正も量子化もなければ、図3の出力

【0067】

【外3】

$\hat{X}(Z)$

【0068】は次式により与えられる。

【0069】

【数1】

$$\hat{X}(Z) = \frac{1}{2} [H_0(Z)G_0(Z) + H_1(Z)G_1(Z)]X(Z) + \frac{1}{2} [H_0(-Z)G_0(Z) + H_1(-Z)G_1(Z)]X(-Z)$$

【0070】本発明において、

【0071】

【外4】

$\hat{X}(Z)$

【0072】の第2項は”エイリアシング”（折り返

し) 項と呼ばれるが、これはキャンセルされる。というのは、合成フィルタが分析フィルタの直交ミラーフィルタとされているからである。すなわち、

【0073】

【数2】

$$\begin{cases} G_0(Z) = H_1(-Z) \\ G_1(Z) = -H_0(-Z) \end{cases}$$

【0074】フィルタ係数によれば、

【0075】

【数3】

$$\begin{cases} g_0(n) = (-1)^n h_1(n) \\ g_1(n) = -(-1)^n h_0(n) \end{cases}$$

【0076】したがって、直交ミラーフィルタのペアについて、代入すると出力は次のようになる。

【0077】

【数4】

$$\hat{X}(z) = \frac{1}{2} [H_0(Z)H_1(-Z) - H_1(Z)H_0(-Z)]X(Z)$$

【0078】よって、本発明の直交ミラーシステムにおいて、出力は分析フィルタの項だけで決定される。ウェーブレット変換は、フィルタにより生成された出力がフィルタの入力として直接的または間接的に用いられるという点で、変換信号に対し再帰的に適用される。ここに述べる実施例においては、低域通過変換成分 $y_0(n)$ だけが再帰的に変換されるため、当該システムはピラミッド型である。そのようなピラミッドシステムの一例が図10に示されている。

【0079】Z変換はハードウェア及び/またはソフトウェアのデータ操作の説明に便利な表現方法である。Zによる乗算は、ハードウェアによるmクロックサイクルの遅延、及び、ソフトウェアによるm個前の要素への配列アクセスのモデルである。そのようなハードウェア手段は、メモリ、パイプステージ、シフタ、レジスタ等を含む。

【0080】本発明において、信号 $x(n)$ と信号

【0081】

【外5】

$\hat{x}(n)$

【0082】は、ある乗定数及びある遅延期間までは一致する。すなわち、Z変換では

【0083】

【数5】

$$\hat{X}(Z) = cZ^{-m}X(Z)$$

【0084】これは精密復元システムと呼ばれる。しかし、本発明の一実施例において、入力データに最初に

適用されたウェーブレット変換は精密復元可能である。

【0085】ハダマード変換 (Hadamard Transform) を使う本発明の一実施例は精密復元システムであり、その正規化形はZ領域で次のように記述される。

【0086】

【数6】

$$\begin{cases} H_0(Z) = \frac{1}{\sqrt{2}}(1+Z^{-1}) \\ H_1(Z) = \frac{1}{\sqrt{2}}(1-Z^{-1}) \end{cases}$$

【0087】代入すると、出力は次のとおりである。

【0088】

【数7】

$$\hat{X}(Z) = Z^{-1}X(Z)$$

【0089】これが精密復元であることは明らかである。ハダマード変換のこれ以上の情報を得るには、Anil K. Jain, Fundamentals of Image Processing, p. 155を見られたい。

【0090】ハダマード変換の可逆バージョンは本明細書においてS変換と呼ばれる。

【0091】S変換に関する以上の情報を得るには、Said, A. and Pearlman, W., "Reversible Image Compression via Multiresolution Representation and Predictive Coding", Dept. of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 1993 を参照されたい。ハダマード変換は精密復元変換であるから、下に示す非正規化バージョン（定数因子がハダマード変換と異なる）も精密復元変換である。

【0092】

【数8】

$$\begin{cases} h_0(Z) = \frac{1}{2}(1+Z^{-1}) \\ h_1(Z) = 1-Z^{-1} \end{cases}$$

【0093】入力信号のサンプルを x_0, x_1 とすると、S変換は当該システムの可逆システムである。すなわち、

【0094】

【数9】

$$\begin{cases} y_0(0) = \lfloor (x(0)+x(1))/2 \rfloor \\ y_1(0) = x(0) - x(1) \end{cases}$$

【0095】S変換は、生成インデックス n を用いて出力を次のように定義することができる。

【0096】

【数10】

$$\begin{cases} s(n) = \left\lfloor \frac{x(2n) + x(2n+1)}{2} \right\rfloor \\ d(n) = x(2n) - X(2n+1) \end{cases}$$

【0097】なお、変換係数アドレッシングの因数2は、暗黙の2:1サブサンプリングの結果である。この変換は可逆であり、その逆変換は次の通りである。

【0098】

【数11】

$$\begin{cases} x(2n) = s(n) + \left\lfloor \frac{d(n)+1}{2} \right\rfloor \\ x(2n+1) = s(n) - \left\lfloor \frac{d(n)}{2} \right\rfloor \end{cases}$$

【0099】記号

【0100】

【外6】

└.┘

【0101】は、切り捨てる意味を意味し、時に床関数と呼ばれる。同様に、天井関数

【0102】

【外7】

⌈.⌉

【0103】は最も近い整数へ切り上げることを意味する。

【0104】このシステムが可逆であることは、近似により失われる情報が $x(0) + x(1)$ の最下位ビットだけで

$$\begin{cases} H_0(Z) = \frac{1}{\sqrt{2}}(1+Z^{-1}) \\ H_1(Z) = \frac{1}{8\sqrt{2}}(-1-Z^{-1}+8Z^{-2}-8Z^{-3}+Z^{-4}+Z^{-5}) \end{cases}$$

【0109】代入すると、出力は

【0110】

【数14】

$$\hat{X}(Z) = 2Z^{-3}X(Z)$$

$$\begin{cases} h_0(Z) = \frac{1}{2}(1+Z^{-1}) \\ h_1(Z) = \frac{1}{8}(-1-Z^{-1}+8Z^{-2}-8Z^{-3}+Z^{-4}+Z^{-5}) \end{cases}$$

【0113】 $x(0), x(1), \dots, x(5)$ が信号の6サンプルであるとき、初めの3つの低域通過係数 $y_0(0), y_0(1), y_0(2)$ と最初の高域通過係数 $y_1(0)$ は、次式で与えられる。

【0114】

あるという事実から証明される。しかしながら、 $x(0) + x(1)$ と $x(0) - x(1)$ の最下位ビットは同一であるので、これは高域通過出力 $y_1(0)$ から再生することができる。つまり、

【0105】

【数12】

$$\begin{cases} x(0) = y_0(0) + \lceil (y_1(0)+1)/2 \rceil \\ x(1) = y_0(0) - \lceil (y_1(0)-1)/2 \rceil \end{cases}$$

【0106】S変換は、最小長 (minimal length) の可逆フィルタを用いる非オーバーラップ (non-overlapping) 変換である。最小長フィルタは、1対の2タップ・フィルタからなる。最小長変換は良好なエネルギー集中を得られない。最小長フィルタは、その長さがフィルタ数に等しいので、非オーバーラップ変換を実現する。オーバーラップ変換は、フィルタ数より長さが大きいフィルタを少なくとも1つ用いる。長い（非最小長の）フィルタを使うオーバーラップ変換は、より良好なエネルギー集中を与えることができる。本発明は、オーバーラップ変換を可能にする非最小長の可逆フィルタを提供する。

【0107】精密復元システムのもう一つの例は、次のZ領域定義を持つTwo/Six (T/S) 変換からなる。

【0108】

【数13】

【0111】であり、これは精密復元変換である。このT/S変換の有理非正規化バージョンは次のとおりである。

【0112】

【数15】

【数16】

$$\begin{cases} y_0(0) = \lfloor (x(0)+x(1))/2 \rfloor \\ y_0(1) = \lfloor (x(2)+x(3))/2 \rfloor \\ y_0(2) = \lfloor (x(4)+x(5))/2 \rfloor \\ y_1(0) = \lfloor [-(x(0)+x(1))] + 8(x(2)-x(3)) + (x(4)+x(5))/8 \rfloor \end{cases}$$

【0115】しかし、このT S変換の有理非正規化バージョンをそのまま具体化したのでは可逆でない。次に述べる例で、そのような構成が局所的に非可逆であることが明らかになる。グローバルケースのための例としては、もっと長い系列を作ることができる。 $y_0(0)$ と $y_0(0)$ と

$$\begin{aligned} y_0(0) &= \lfloor (1+1)/2 \rfloor = 1 \\ y_0(1) &= \lfloor (3+1)/2 \rfloor = 2 \\ y_0(2) &= \lfloor (1+1)/2 \rfloor = 1 \\ y_1(0) &= \lfloor [-(1+1) + 8(3-1) + (1+1)]/8 \rfloor = \lfloor (-2+16+2)/8 \rfloor = 2 \end{aligned}$$

【0118】また、 $x(0)=1, x(1)=2, x(2)=4, x(3)=1, x(4)=1, x(5)=1$ であるときは、

$$\begin{aligned} y_0(0) &= \lfloor (1+2)/2 \rfloor = 1 \\ y_0(1) &= \lfloor (4+1)/2 \rfloor = 2 \\ y_0(2) &= \lfloor (1+1)/2 \rfloor = 1 \\ y_1(0) &= \lfloor [-(1+2) + 8(4-1) + (1+1)]/8 \rfloor = \lfloor (-3+24+2)/8 \rfloor = \lfloor 23/8 \rfloor = 2 \end{aligned}$$

【0120】 $y_0(0), y_0(1), y_1(0)$ は異なった2組の入力 $x(0) \dots x(5)$ に対して同一であるから、この変換は可逆でない。 $y_0(0), \dots, y_1(0)$ を与えられたとき、このローカル情報から、その2つの組のいずれが入力されたのか判断できないからである。(ただし、全係数から得られるグローバル情報を用いるときに、当該変換が可逆でないとは証明できない。)

さて、異なった高域通過フィルタ作用を与える可逆T S

$$\begin{cases} y_0(0) = \lfloor (x(0)+x(1))/2 \rfloor \\ y_0(1) = \lfloor (x(2)+x(3))/2 \rfloor \\ y_0(2) = \lfloor (x(4)+x(5))/2 \rfloor \end{cases}$$

$$\begin{aligned} y_1(0) &= \lfloor [-(x(0)+x(1))/2] + 4(x(2)-x(3)) + (x(4)+x(5))/2 + 2 \rfloor / 4 \\ &= \lfloor (-y_0(0) + 4(x(2)-x(3)) + y_0(2) + 2) / 4 \rfloor \end{aligned}$$

【0123】

【数20】

$$x(2)-x(3) = y_1(0) - \lfloor (-y_0(0) - y_0(2) + 2) / 4 \rfloor$$

【0124】であるので、 $x(2)-x(3)$ は完全に既知となった。前記(数20)の $y_0(1)$ と、 $x(2)-x(3)$ と $x(2)-x(3)$ が上のように決まれば、 $x(0)+x(1)$ と $x(0)-x(1)$ の最下位ビットは同一であるから、 $x(2)$ と $x(3)$ を復元できる。つまり、次のよう仮定する。

【0125】

【数21】

(2)を計算するために丸めを行なうため、

$$-(x(0)+x(1)) + (x(4)+x(5)) \neq -y_0(0) + y_0(2)$$

であるから、ローカル情報を使うとき当該変換は可逆でない。

【0116】例えば、 $x(0)=1, x(1)=1, x(2)=3, x(3)=1, x(4)=1, x(5)=1$ であるときは、

【0117】

【数17】

【0119】

【数18】

変換を考えよう。この変換を、ここではR T S変換と呼ぶ。

【0121】 $x(0), x(1), x(2), x(3), x(4), x(5)$ が信号の6サンプルであるとき、初めの3つの低域通過係数 $y_0(0), y_0(1), y_0(2)$ と最初の高域通過係数 $y_1(0)$ は次式で与えられる。

【0122】

【数19】

$$\begin{aligned} d(0) &= x(2)-x(3) = y_1(0) - \lfloor (-y_0(0) + y_0(2) + 2) / 4 \rfloor \\ x(2) &= y_0(1) + \lfloor (d(0)+1)/2 \rfloor \\ x(3) &= y_0(1) + \lceil (d(0)-1)/2 \rceil \end{aligned}$$

【0126】R T S変換の一実施例並びにS変換の一実施例においては、8による除算は、精度を上げるために、2で割ってから4で割ることによって行われる。なお、数学的には次の2式

【0127】

【数22】

23

$$\frac{1}{8}(-1 - Z^{-1} + 8Z^{-2} - 8Z^{-3} + Z^{-4} + Z^{-5} + 4)$$

【0128】

【数23】

$$\frac{1}{4} \left(\frac{1}{2}(-1 - Z^{-1}) + 4(Z^{-2} - Z^{-3}) + \frac{1}{2}(Z^{-4} + Z^{-5}) + 2 \right)$$

【0129】は、無限精度演算による時には同一である。(数23)の式が可逆フィルタを表わしている理由は、整数演算で実際に行ってみれば明白である。低域通過フィルタと高域通過フィルタをハードウェアにより実現した典型例は、図21及び図22に関連して説明する。

【0130】なお、S変換及びRTS変換の両方において、低域通過フィルタは入力信号 $x(n)$ のレンジが出力信号 $y_0(n)$ のレンジと同一になるように構成される。例

$$\begin{cases} s(n) = \left\lfloor \frac{x(2n) + x(2n+1)}{2} \right\rfloor \\ d(n) = \left\lfloor \frac{\frac{x(2n) + x(2n+1)}{2} + 4(x(2n+2) - x(2n+3)) + \frac{x(2n+4) + x(2n+5)}{2}}{4} \right\rfloor \end{cases}$$

【0133】この $d(n)$ の式は $s(n)$ を使って単純化することができる(さらに、4による整数除算は分子に2を加算することにより丸めることができる)。これにより次式が得られる。

【0134】

【数25】

$$\begin{cases} s(n) = \left\lfloor \frac{x(2n) + x(2n+1)}{2} \right\rfloor \\ d(n) = x(2n+2) - (2n+3) + \left\lfloor \frac{-s(n) + s(n+2) + 2}{4} \right\rfloor \end{cases}$$

【0135】このTS変換は可逆であり、また、その逆変換は次式である。

【0136】

【数26】

$$\begin{cases} x(2n) = s(n) + \left\lfloor \frac{p(n)+1}{2} \right\rfloor \\ x(2n+1) = s(n) - \left\lfloor \frac{p(n)}{2} \right\rfloor \end{cases}$$

【0137】ここで、 $p(n)$ は次式により最初に計算されなければならない。

【0138】

【数27】

$$p(n) = d(n-1) - \left\lfloor \frac{-s(n-1) + s(n+1) + 2}{4} \right\rfloor$$

【0139】低域通過フィルタにより得られる結果を高

24

れば、信号が8ビットの画像であるときには、低域通過フィルタの出力も8ビットである。このことは、低域通過フィルタが連続的に適用されるピラミッドシステムにとつては重要な特性である。というのは、従来システムでは、出力信号のレンジが入力信号のレンジより大きく、そのことがフィルタの連続的適用を難しくしていたからである。さらに、低域通過フィルタは2つのタップしか持たないため、フィルタが非オーバーラップフィルタになる。この性質は、後述のように、フィルタをハードウェアで実現する上で重要である。

【0131】より一般的な形で表すならば、可逆TS変換は、低域通過フィルタと高域通過フィルタの2出力の式により定義される。すなわち

【0132】

【数24】

域通過フィルタにおいて2度(第1項と第2項で)利用できる。したがって、高域通過フィルタの結果を得るには、ほかに2つの加算を行うだけでよい。

【0140】TS変換は、可逆であるほかに、効率的である。よって、TS変換は非損失性圧縮に非常に適している。TS変換は(S変換と同様に)、平滑出力の増大が全くない、すなわち、入力信号が b ビットの深さのとき、平滑出力も b ビットの深さである。このことは、平滑出力がさらに分割される後述のピラミッドシステムの場合に有益である。変換を整数演算で行う際の丸めによる系統誤差はないので、損失性システムの全ての誤差を量子化により制御可能である。

【0141】ウェーブレット変換に関わる4つのフィルタの中で、低域通過合成フィルタが最も重要である。なぜなら、それは量子化された係数を結合し、さらに、その結果を平滑化するからである。このことから、本発明にあっては、比較的長い(6タップ)特に性能のよいフィルタを低域通過合成フィルタとして選択するに至った。なお、QMFシステムでは、独立したフィルタは2個しかない。

【0142】多くのオーバーラップ、非最小長の可逆フィルタを本発明に用いることができる。非オーバーラップ最小長可逆フィルタを使ってフィルタリングを行なう変換システムのフォワード変換とインバース変換の説明を図4に示す。例えば、次に示す種類のフィルタを本発明で使ってもよい。ただし、整数 $L \geq z$ とする。

【0143】

【数28】

$$d(0) = x(2(\lfloor L/2 \rfloor + 1)) - x(2(\lfloor L/2 \rfloor + 1) + 1)$$

$$y_0(0) = \lfloor (x(0) + x(1)) / 2 \rfloor$$

$$y_0(1) = \lfloor (x(2) + x(3)) / 2 \rfloor$$

$$y_0(L-1) = \lfloor (x(2(\lfloor L/2 \rfloor + 1)) + x(2(\lfloor L/2 \rfloor + 1) + 1)) / 2 \rfloor$$

【0145】

【数30】

$$y_1(0) = \frac{\sum_{i=0}^{\lfloor L/2 \rfloor} a_i y_0(i) + b d(0) + \sum_{j=\lfloor L/2 \rfloor + 2}^{L-1} c_j y_0(j) + \lfloor \frac{k}{2} \rfloor}{k}$$

【0146】この高域通過フィルタの長さは $2L$ である。 L が奇数ならば、このフィルタはより対称フィルタに近いものになろう。 a_i, b, c_i, k が整数で、 $k \leq b$ のときは、このフィルタは可逆である。 a_i, b, c_i, k が 2 のべき乗（あるいは、 2 のべき乗の負値もしくは補数）であれば、フィルタの構成を簡略化でき

$$\sum_{i=0}^{\lfloor L/2 \rfloor} [(a_i)(2i)^m + (a_i)(2i+1)^m] + (b)(2(\lfloor L/2 \rfloor + 1))^m - (b)(2(\lfloor L/2 \rfloor + 1) + 1)^m + \sum_{j=\lfloor L/2 \rfloor + 2}^{L-1} [c_j(2j)^m + c_j(2j+1)^m] = 0$$

【0150】この性質によって、高域通過フィルタは、 $m=1$ の時に線形変化する入力に対し応答しなくなり、 $m=2$ の時に二次曲線的に変化する入力信号に対し応答しなくなる、等々となる。ここで m はモーメント条件である。この性質が、RTS変換がS変換よりエネルギー集中がよいということの主な根拠である。

【0151】フィルタは可逆性のための最低条件を充足しなければならないけれども、用途によって、他の特性

$$\begin{array}{cccccccccccc} 1 & 1 & -4 & -4 & 16 & -16 & 4 & 4 & -1 & -1 \\ 1 & 1 & -3 & -3 & 8 & -8 & 3 & 3 & -1 & -1 \\ -1 & -1 & 0 & 0 & 16 & -16 & 0 & 0 & 1 & 1 \\ -1 & -1 & 4 & 4 & -16 & -16 & 256 & -256 & 16 & 16 & -4 & -4 & 1 & 1 \\ 3 & 3 & -22 & -22 & 128 & -128 & 22 & 22 & -3 & -3 \end{array}$$

最後のフィルタは(Two/Ten) TT-フィルタであり、3次増加関数に対して応答しないという性質を有する。なお、 $22=16+2\times 3$ 、 $3=2+1$ であるので、このフィルタは合計7つの加減算によって実現できることに注意されたい。

【0153】一実施例においては、複数のフィルタが結合されて1つのブロックに適用されることにより、水平パスと垂直パスの両方が1操作で実行される。図5は、そのような結合動作を行うフィルタを示している。構成(a)は、1-Dフィルタと1-D丸め操作からなる1-D可逆フィルタを各パス毎に1個ずつ、計2個使用することを表している。構成(b)は1-Dフィルタ401の後に別の1-Dフィルタ402が続き、2-D丸め

【0144】
【数29】

る。

【0147】(a_i, c_i の値と関係なく) $k = b$ ならば、高域通過フィルタ出力 y_1 のレンジは最小になる。各 a_i について、 $a_i = -c_i$ となる c_i が丁度1つ存在するときには、高域通過フィルタは一定した入力に対し全く応答しない。 $j - (L-1) = i$ の時に $a_i = -c_i$ ならば、フィルタは、対称フィルタにより近いものになる。

【0148】もう一つの有用な性質は次のとおりである。

【0149】

【数31】

のどれも満たさないフィルタ、いくつかを満たすフィルタ、あるいは全部を満たすフィルタを用いることができる。いくつかの実施例においては、次に示す高域通過フィルタの1つが用いられる。本発明を難解にしないため、これらのフィルタは、フィルタの有理バージョンの整数係数が単に表形式で示されている。

【0152】

$$\begin{array}{cccccccccccc} 1 & 1 & -4 & -4 & 16 & -16 & 4 & 4 & -1 & -1 \\ 1 & 1 & -3 & -3 & 8 & -8 & 3 & 3 & -1 & -1 \\ -1 & -1 & 0 & 0 & 16 & -16 & 0 & 0 & 1 & 1 \\ -1 & -1 & 4 & 4 & -16 & -16 & 256 & -256 & 16 & 16 & -4 & -4 & 1 & 1 \\ 3 & 3 & -22 & -22 & 128 & -128 & 22 & 22 & -3 & -3 \end{array}$$

操作403で終わることを表している。この構成は、より良好な丸めが可能であるという点でより精密な結果を生じる。

【0154】フィルタの厳密な可逆性要件は、次のこと留意することにより緩和できる。高域通過係数は、ある順序で符号化され、そして復号される。既に復号された高域通過係数に対応する画素値は正確に分かれているから、それを現在の高域通過フィルタリングに利用できる。例えば、ラスター順が用いられる時、下記のフィルタを使用できる。

【0155】

【数32】

$$H_1(Z) = \left[\frac{1}{4} \left(\left[-\frac{1}{2} (1 + Z^{-1}) \right] + \left[\frac{1}{2} (8(Z^{-2} - Z^{-3}) + (Z^{-4} + Z^{-5})) \right] + 2 \right) \right]$$

【0156】单一の固定高域通過フィルタを使わねばならないというわけではない。適応フィルタを使ってもよいし、あるいは複数のフィルタを使うこともできる。複数のフィルタの適応化ないし選択のために利用するデータは、特定のインバースフィルタリング動作に先立って復号器内で入手可能なデータに限定されなければならぬ。

【0157】複数のフィルタを使用する一方法は、高域通過係数をプログレッシブに処理することである。1つおきの高域通過フィルタ演算 ($y_1(0)$, $y_1(2)$, $y_1(4)$, ...) は、RTS高域通過フィルタのような可逆フィルタを用いて最初に処理してよい。残りのフィルタ操作 ($y_1(1)$, $y_1(3)$, $y_1(5)$, ...) には、最高6タップの非可逆フィルタを使用できる。というのは、フィルタのオーバーラップ部分への入力の正確な値が分かっているからである。例えば、次に示すフィルタのどれを使用してもよい。

【0158】

-1	3	-3	1		
-1	4	-4	1		
-3	8	-8	3		
1	-5	10	-10	5	-1
1	-4	8	-8	4	-1

なお、実施例によっては、QMFフィルタは使われない。

【0159】実施例によっては、高域通過フィルタは予測／補間（内挿）操作に置き換えられるかもしれない。予測器／補間器は、特定の予測／補間操作の前に、復号器で入手可能なデータを利用し、1対の入力間の差分を予測する。予測した差分が入力の実際の差分から減算され、その結果が出力される。一実施例では、DPCM、プログレッシブ符号化あるいは空間領域符号化に使われている従来の予測方法が用いられる。

【0160】一実施例では、形態学的(morphological)フィルタ（例えばメジアンフィルタ）のような非線形フィルタが用いられるかもしれない。一実施例では、1, 1フィルタが、高域通過用の別のフィルタと共に用いられる。そのようなフィルタ系は2画素間の差を伝達できなければならない。復号器の有する任意のデータに基づき、その差がどうなるか予測することができる。この予測をするために、非線形形態学的フィルタを利用できる。本発明は、そのためにウインドウの原因側(casual side)の実画素を利用し、それをフィルタに入力することにより画素周りのメジアンを計算する。フィルタの非原因側では、低域通過係数が画素値の代わりに利用される。

【0161】<2次元ウェーブレット分割>本発明の低

域通過フィルタ及び高域通過フィルタを用いて、マルチ解像度分割が行なわれる。分割レベル数は可変であり任意数でよいが、現在のところ分割レベル数は2乃至5レベルである。

【0162】画像のような2次元データに対する変換を実行する最も普通のやり方は、1次元フィルタを別々に適用する方法、つまり、行に沿って適用したのち列に沿って適用するという方法である。第1レベルの分割により4つの異なる係数バンド（ここではLL, HL, LH, HHと呼ぶ）が得られる。それら文字は、前に定義した平滑フィルタと細密フィルタの適用を意味する低（L）と高（H）をそれぞれ表す。したがって、LLバンドは平滑フィルタから得られる行方向及び列方向の係数からなっている。ウェーブレット係数を図6乃至図9のような形に配置するのが一般的なやりかたである。

【0163】ウェーブレット分割の各サブブロックをさらに分割することができる。最も普通のやりかたはLLサブブロックだけをさらに分割する方法であり、これは若干数繰り返して行うことができる。そのような多重分割はピラミッド分割と呼ばれる（図6乃至図9）。記号LL, LH, HL, HHと分割レベル番号によって各分割を示す。なお、本発明のSフィルタ、TSフィルタのいずれの場合も、ピラミッド分割は係数サイズを増加させない。

【0164】例えば、可逆ウェーブレット変換が再帰的に1つの画像に適用されるときには、第1レベルの分割は最も細かいディテールもしくは解像度に対し作用する。第1分割レベルでは、画像は4つのサブ画像（すなわちサブバンド）に分割される。各サブバンドは、1つの空間周波数帯域を表わしている。第1レベルのサブバンドはLL0, LH0, HL0, HH0と表される。原画像を分割するプロセスは、水平、垂直の両次元の2:1のサブサンプリングを含むので、図6に示されるように、第1レベルのサブバンドLL0, LH0, HL0, HH0はそれぞれ、入力が持っている画像の画素（または係数）の個数の4分の1の個数の係数を持っている。

【0165】サブバンドLL0は、水平方向の低い周波数情報と垂直方向の低い周波数情報を同時に含んでいる。一般に、画像エネルギーの大部分は当該サブバンドに集められる。サブバンドLH0は、水平方向の低い周波数情報と垂直方向の高い周波数情報（例えば水平方向エッジ情報）を含んでいる。サブバンドHL0は、水平方向の高い周波数情報と垂直方向の低い周波数情報（例えば垂直方向エッジ情報）を含んでいる。サブバンドHH0は、水平方向の高い周波数情報と垂直方向の高い周波数情報（例えばテクスチャもしくは斜めエッジ情報）を含んでいる。

【0166】この後に続く第2、第3、第4の下位分割レベルはそれぞれ、前レベルの低周波数LLサブバンドを分割することによって作られる。第1レベルの当該サブバンドLL0が分割されることによって、やや精細な第2レベルのサブバンドLL1, LH1, HL1, HH1が作られる(図7)。同様に、サブバンドLL1が分割されることによって、精細度の粗い第3レベルのサブバンドLL2, LH2, HL2, HH2が生成される(図8)。また、図9に示されるように、サブバンドLL2が分割されることにより、精細度がより粗い第4レベルのサブバンドLL3, LH3, HL3, HH3が作られる。2:1のサブサンプリングにより、第2レベルの各サブバンドは、原画像の1/6分の1の大きさである。このレベルの各サンプル(つまり画素)は、原画像中の同一位置のやや細いディテールに相当する。同様に、第3レベルの各サブバンドは、原画像の1/4分の1の大きさである。第3レベルでの各画素は、原画像中の同一位置のかなり粗いディテールに相当する。また、第4レベルの各サブバンドは、原画像の1/256分の1の大きさである。

【0167】分割された画像はサブサンプリングのため原画像より物理的に小さいので、原画像の格納のために使用されるメモリを利用して、分割サブバンド全部を格納できる。つまり、3レベル分割の場合、原画像と分割サブバンドLL0, LL1は捨てられ、保存されない。

【0168】4つのサブバンド分割レベルだけを示したが、個々のシステムの条件に応じて、それ以上のレベルを生成することも可能である。また、DCTのような他の変換又は一次元配置のサブバンドによって様々な親子関係が定義されてもよい。

【0169】ピラミッド分割のウェーブレット係数には自然で有益なツリー構造がある。なお、最終分割レベルに対応するLLサブブロックはただ一つである。これに対し、レベル数と同数のLH, HL, HHブロックがある。このツリー構造により、ある周波数帯域内の係数の親は、それより低解像度の同じ周波数帯域内の係数であり、かつ同じ空間位置関係にあることが明らかになる。図10は連続した2レベル間の親子関係を表している。

【0170】図10において、Aの係数は、B, C, Dに対する直接の親であるが、またさらに、B, C, Dを親とする係数に対する親もある。つまり、Bは、E付近の4係数に対する親である、またH付近の16係数に対する親である、等々である。

【0171】マルチ解像度分割のプロセスは、図11に示すようなフィルタ系列を使って行なうことができる。長さLの一次元信号を表わす入力信号は、フィルタユニット701, 702により低域通過フィルタ処理及び高域通過フィルタ処理を施され後、ユニット703, 704により2:1のサブサンプリングを施される。ユニット703によりサブサンプリングされた出力信号は、ユニット705, 706により低域通過フィルタ処理及び

高域通過フィルタ処理を施された後、ユニット707, 708により2:1のサブサンプリングを施される。サブバンド成分L, Hがユニット707, 708の各出力に得られる。同様に、ユニット709, 710により低域通過フィルタ処理及び高域通過フィルタ処理を施された後、ユニット711, 712によりそれぞれサブサンプリングされる。サブバンド成分L, Hがユニット711, 712の各出力に得られる。上に述べたように、本発明の一実施例においてサブバンド分割に用いられるフィルタは、水平周波数帯域及び垂直周波数帯域を低周波数帯域及び高周波数帯域へ分割するためのデジタル直交ミラーフィルタである。

【0172】図12は、2次元2レベル変換を示している。図13も、図21及び図22に示したような1次元フィルタを使って実現した2次元2レベル変換を示している。それら1次元フィルタ761～766は、サブサンプリングにより不要になる演算を避けるため1画素位置おきに適用される。一実施例では、1次元フィルタは低域通過フィルタ演算と高域通過フィルタ演算との間で演算を共有する。

【0173】したがって、本発明は、非最小長のオーバーラップ可逆フィルタが使用された圧縮・伸長システムを提供する。図14は、そのようなシステムの一実施例のブロック図である。図14において、最初に分割プロック1001で階層的分割が行なわれる。この階層的分割の結果は圧縮器1002へ送られて圧縮される。ここで実行される圧縮には、ベクトル量子化、スカラー量子化、ゼロ・ランレンジス符号化、ハフマン符号化、Tunstall等々が含まれよう。圧縮器1002の出力は、オリジナル入力データを圧縮したものを表わすデータである。伸長器1003は、そのデータを、いつか受け取つて伸長することになろう。その時に、本発明は、逆分割プロック1004において非最小長オーバーラップ可逆フィルタを用い逆分割を行なって、オリジナルデータの復元データを生成する。なお、この非最小長オーバーラップ可逆フィルタは非S変換フィルタである。

【0174】本発明の可逆ウェーブレットフィルタは、図15に示すような典型的な分析及び強調システムにも使用し得る。図15において、非最小長オーバーラップ可逆ウェーブレットフィルタ1101を使って、入力データに対し階層的分割が行なわれる。分析ユニット1102は、フィルタにより生成された係数を受け取り、それら係数をデシジョンに分類する。すなわち、係数を完全に符号化するのではなく、当面関係のある情報だけが抽出される。例えば、文書アーカイブシステムにおいて、空白のページは最も粗い低域通過サブバンドだけを利用して認識できるであろう。もう一つ例を挙げれば、ある特定のサブバンドの高域通過情報だけを利用して、テキスト画像と自然シーン画像とを区別する。階層的分割は、最初に粗いサブバンドによって粗いレジストレー

ション (registration) が行なわれるような、複数画像のレジストレーションにも利用できる。他の実施例においては、係数はユニット 1106 で強調処理またはフィルタ処理を施されてからユニット 1107 で逆分割が行なわれる。階層分割を利用して鮮明化、エッジ強調、ノイズ抑制等々を行なうこともできる。このように、本発明は、統合時間／空間領域及び周波数領域の分析システム並びにフィルタ処理／強調処理システム用のウェーブレット変換を提供する。

【0175】<係数及びビットプレーンの順序付け及びモデリング>本発明では、ウェーブレット分割の結果として生成される係数はエントロピー符号化される。本発明においては、係数は最初に埋め込み符号化 (embedded coding) を施されるが、この符号化では、視覚的に重要な順に係数が順序付けられ、また、より一般的には何等かの誤差規準 (例えば、歪み規準) を考慮して係数が順序付けられる。誤差または歪みの規準には、ピーク誤差と平均 2 乗誤差 (MSE) が含まれる。なお、ビット有意度空間配置 (bit-significance spatial location) より、データベース照会のための妥当性を優先させるように、また方向別に (垂直、水平、斜め等)、順序付けをすることができる。

【0176】データの順序付けが行われることにより、符号ストリームの埋め込み量子化したものを生成する。本発明においては、2 つの順序付け方法が利用される。その一つは係数を順序付けするためのものであり、もう一つは係数中の 2 進値を順序付けするためのものである。本発明の順序付けはビットストリームを生成し、このビットストリームはその後に 2 元エントロピー符号化器により符号化される。

【0177】一実施例では、係数の符号化及びモデリングは M 元符号化を構成する。別の実施例では、ビット単位でなく、バンド単位でのみ埋め込まれる。また、非損失性符号化又はシングルクオリティ (single quality) 損失性符号化 (例えば符号化器で規定される量子化) のために、非埋め込み符号化が係数の順序付け及びモデリングと一緒に利用されてもよい。

【0178】<符号化単位>本発明において、符号化単位はツリーの矩形集合であり、これは画像の残りの部分と独立に符号化される。符号化単位は符号化データの最小単位を表す (符号化単位の部分的な符号化を許す量子化オプションもあるが)。1 符号化単位内のデータを全部、符号化器は一度に入手でき、例えばメモリにバッファされる。

【0179】符号化単位の選び方は構成次第である。符号化単位は、画像全体 (もしくは他のデータ集合) 又は本発明の 1 つのツリー、あるいはその中間の矩形と定義されるかもしれない。一実施例では、符号化単位の選択は圧縮効率とメモリ使用量との間の兼ね合いで決まることになろう。

【0180】一実施例においては、符号化単位内の全ての係数がランダムアクセスメモリ内にある。符号化単位内の全係数がランダムアクセスメモリ内にあるから、符号化単位内の係数間の埋め込み順序を任意の順序にできる。この順序は符号化器と復号器の両方が知っている。しかしエントロピー符号化器はこの順序付けに関して原因側であるから、その順序は圧縮に重大な影響を及ぼすので慎重に選ばれる。順序付けの一実施例を後に述べる。

【0181】<モデリング>本発明においては、統合空間／周波数モデリングが、本発明のウェーブレット変換により生成された係数の符号化に利用される埋め込み符号化システムを構成する。統合空間／周波数モデリングは、既知の周波数帯域と近傍画素 (またはデータ) を両方利用する。統合空間／周波数モデリングの一実施例は、本明細書において水平 (horizon) モデリングと呼ばれる。

【0182】データはまず符号／絶対値形式にフォーマットされ、次に該データは有意度に基づいて並べ替えられる。別の実施例では、作業領域メモリをさらに減らすため、係数は符号／絶対値形式でなく絶対値／化数形式で格納してもよい。

【0183】データは、所与の有意度規準に関連して並べ替えられた後、符号化される。

【0184】ある信号 $x(n)$ を想定し、各 $x(n)$ が R ビットの精度で表現されているとした場合、本発明の埋め込み符号化は、その信号の各 $x(n)$ の最上位の 1 ビット (または複数ビット) を符号化し、次にその下位の 1 ビット (または複数ビット) を符号化し、さらに下位のビットというように符号化する。例えば、視覚的に定義される順序付けの場合、中央部分でコーナー沿いまたは縁近傍より高い品質を要求する画像 (例えば、ある種の医用画像) は、中央部分の画素の下位ビットが周辺部画素の上位ビットより先に符号化されるような符号化を施されよう。

【0185】<ビット有意度 (Bit-Significance) 表現>一実施例では、係数内の 2 進値に対し用いられる埋め込み順序はビットプレーン順である。係数はビット有意度表現で表される。ビット有意度は、最上位ビット (MSB) ではなくて、符号ビットが最初の非ゼロ絶対値ビットと共に符号化される符号／絶対値表現である。

【0186】ビット有意度形式で表現される数には 3 種類のビット、すなわちヘッド (head) ビット、テール (tail) ビット及び符号 (sign) ビットがある。ヘッドビットとは、MSB から最初の非ゼロ絶対値ビットまでの全てのゼロビットに、その最初の非ゼロ絶対値ビットを含めたものである。その最初の非ゼロ絶対値ビットが存在するビットプレーンで、係数の有意度が定まる。最初の非ゼロ絶対値ビットの後から LSB までのビットがテールビットである。符号ビットは符号 (sign) を表示するにすぎ

ない。MSBが非ゼロビットの数は、ヘッドビットが1ビットだけである。ゼロの係数は、テールビットも符号ビットも持たない。

【0187】画素の輝度に関連して起こるような、値が非負の整数の場合、採用し得る順序はビットプレーン順（例えば、最上位ビットプレーンから最下位ビットプレーンへの順）である。2の補数による負整数も許容される実施例では、符号ビットの埋め込み順序は、整数の絶対値の最初の非ゼロビットと同じである。したがって、1つの非ゼロビットが符号化されるまで、符号ビットは考慮されない。例えば、符号／絶対値表記法によれば、-7の16ビット数は

1000000000000111

である。ビットプレーン・ベースで、初めの12デシジョン(decision)は”非有意”(insignificant)すなわちゼロとなる。最初の1ビットは13番目のデシジョンに存在する。次に、符号ビット（”負”）が符号化される。符号ビットが符号化された後、テールビットが処理される。14番目と15番目のデシジョンは共に”1”である。

【0188】<係数アラインメント(alignment)> FFTやDCTと同様、異なるサブブロック内の係数は異

係数アラインメントの例、b = 8	
1-HH	基準
	10 9 8 7 6 5 4 3 2 1 0
x-HL or x-LH	左3
x-HL or x-LH	左2
x-HL or x-LH	左1
x-HL or x-LH	無
x-HL or x-LH	右1
	8 7 6 5 4 3 2 1 0
	8 7 6 5 4 3 2 1 0
	8 7 6 5 4 3 2 1 0
	8 7 6 5 4 3 2 1 0
	8 7 6 5 4 3 2 1 0

【0192】なお、符号(sign)ビットはMSBではなく、また最初のテールビットと一緒に符号化される。アラインメントはビットがエントロピー符号化器へ送られる順序を制御するにすぎない、ということに注意することが大切である。余分なゼロビットを実際に埋め込んだり、シフトしたり、格納したり、あるいは符号化したりはしない。

【0193】<コンテキストモデル>本発明に利用される水平コンテキストモデルの一実施例を以下に述べる。このモデルは、係数の空間及びスペクトル従属性に基づいて符号化単位内のビットを利用する。隣接した係数及び親係数の利用可能な2進値を使って、コンテキストを生成できる。しかし、コンテキストはデコーダビリティと因果関係があり、また、多少は効率的適応と因果関係がある。

【0194】<エントロピー符号化>一実施例では、本発明により実行されるエントロピー符号化は、2元エントロピー符号化器によって行われる。一実施例では、エ

なった周波数を表す。ビットプレーン符号化の前に、係数を相互に整列させることにより量子化が行われる。量子化の重さが少ない係数ほど早いビットプレーン側へ整列される（例えば左ヘシフトされる）。よって、ストリームが切り捨てられる場合、これらの係数はそれを定義するビットが、それより重く量子化された係数に比べ多くなる。

【0189】一実施例では、係数はSNR又はMSEの見地から最高のレート-歪み性能が得られるように整列される。あるいは、アラインメントは係数データの物理視覚的(physchovisual)量子化を許容する。アラインメントは画像品質の向上に相当な影響を及ぼすが、非損失性圧縮の最終的な圧縮率には殆ど影響しない。

【0190】bビット/画素の入力画像からの2レベルT/S変換分割における幾つかの係数のビット深度(dept h)が図16及び図17に示されている。係数を整列するために、1-HH係数サイズが基準として用いられ、このサイズに対し様々なシフトさせる。表1は、このようなアラインメントプロセスの一例を示している。

【0191】

【表1】

ントロピー符号化器104はQコーダー、QMコーダー、有限状態マシン、高速並列コーダー等である。単一の符号化器を用いて单一の出力符号ストリームを生成してもよい。または、複数の（物理又は仮想）符号化器を用いて複数の（物理又は仮想）データストリームを生成してもよい。

【0195】一実施例では、本発明の2元エントロピー符号化器はQコーダーである。Qコーダーのこれ以上の情報を得るには、Pennebaker, W. B., et al., "An Overview of the Basic Principles of the Q-code Adaptive Binary Arithmetic," IBM Journal of Research and Development, Vol. 32, pg. 717-26, 1988を読まれたい。別の実施例では、2元エントロピー符号化器は、周知の効率的な2元エントロピー符号化器であるQMコーダーを用いる。QMコーダーは、確率スキューリーが非常に高いビットに対して特に効率的である。QMコーダーはJPEG規格とJBIG規格の両方に採用されている。

【0196】2元エントロピー符号化器は、有限状態マシン（FSM）コーダーでもよい。このような符号化器は、確率及び事象（outcome）から圧縮ビットストリームへの単純な変換を提供する。一実施例では、有限状態マシンコーダーは、テーブルルックアップを利用し符号化器及び復号器として実現される。多様な確率予測法を、このような有限状態マシンコーダーに利用できる。確率が0.5に近い場合に圧縮率が非常によい。確率が大きくスキューレーションした場合の圧縮率は、用いられるルックアップテーブルのサイズに依存する。QMコーダーと同様、有限状態マシンコーダーは、デシジョンが発生順に符号化されるので、埋め込みビットストリームに対して有効である。出力はルックアップテーブルにより決められるので、"キャリーオーバー"（carry over）の問題が起こる心配は全くない。実際には、QコーダーやQMコーダーと違って、符号化と圧縮出力ビットの生成までの間に最大遅延時間が存在する。一実施例では、本発明の有限状態マシンコーダーは、1993年12月21日発行の米国特許第5,272,478号 "Method and Apparatus for Entropy Coding" に述べられているBコーダーである。

【0197】一実施例では、本発明の2元エントロピー符号化装置は高速並列コーダーからなる。QMコーダーもFSMコーダーも、1度に1ビットが符号化又は復号されなければならない。高速並列コーダーは、数ビットを並列に処理する。一実施例では、高速並列コーダーは、圧縮性能を犠牲にすることなく、VLSIハードウェア又はマルチプロセッサ・コンピュータで実現される。本発明において利用し得る高速並列コーダーの一実施例は、1995年1月10日発行の米国特許第5,381,145号 "Method and Apparatus for Parallel Decoding and Encoding of Data" に述べられている。

【0198】殆どの効率的なエントロピー符号化器は、基本フィードバックループによって速度が制限される。考えられる一解決法は、入力データストリームを複数のストリームに分割して複数の並列符号化器に与えることである。それら符号化器の出力は、可変長符号化データの複数ストリームである。この種方法の課題の一つは、データを单一のチャンネルでどのようにして伝送するかである。米国特許第5,381,145号に述べられている高速並列コーダーは、この課題を、それら符号化データストリームをインターリーブする方法によって解決する。

【0199】本発明に用いられるコンテキストの多くは固定確率であり、このことがBコーダーのような有限状態マシンコーダーを特に有効なものにする。なお、システムが0.5に近い確率を利用する場合、上記特許に開示された高速並列コーダー及び有限状態マシンコーダーは共にQコーダーより効率よく動作する。よって、それ

ら両方のコーダーは本発明のコンテキストモデルの場合に圧縮上の強みを潜在的に持っている。

【0200】<本発明の符号化及び復号のプロセス>次に述べる図18乃至図20のフローチャートは、本発明の符号化プロセス及び復号プロセスを表している。これらのプロセスは、ソフトウェアで実行されてもハードウェアで実行されてもよい。いずれの場合にも、それを実施する処理ロジックに言及した。

【0201】図18は本発明の符号化プロセスの一実施例を示す。図18において、符号化プロセスはまず処理ロジックに入力データを1符号化単位分だけ取得させる（処理ブロック1301）。つぎに、処理ロジックは、その符号化単位の入力データに可逆フィルタをかける（処理ブロック1302）。

【0202】つぎに、別の分割レベルが必要か判定する（処理ブロック1303）。必要ならば、処理ロジックは全てのLL係数に可逆フィルタをかけ（処理ブロック1304）、そしてプロセスはループバックして処理ブロック1303から処理を継続する。別の分割レベルが必要でなければ、処理ブロック1305から処理を続け、処理ロジックは係数を符号(sign)／絶対値形式に変換する。

【0203】係数を符号／絶対値形式に変換した後、ビットプレーン変数Sが最上位ビットプレーンに設定される（処理ブロック1306）。そして、処理ロジックは随意、エントロピー符号化器を初期化する（処理ブロック1307）。

【0204】エントロピー符号化器が初期化されたならば、処理ロジックは各係数の各ビットをコンテキストモデルでモデリングし、そのビットをエントロピー符号化する（処理ブロック1308）。そのビットをエントロピー符号化した後、データは伝送されるか記憶される（処理ブロック1309）。

【0205】その後、画像中にはかに符号化単位があるか判定する（処理ブロック1310）。ほかに符号化単位があるならば、処理は処理ブロック1301へ続く。一方、ほかに処理単位がなければ、処理は終了する。

【0206】図19は本発明の復号プロセスの一実施例を示す。図19において、プロセスはまず処理ブロックに符号化データを1符号化単位分だけ取り出させる（処理ブロック1401）。次に、変数Sが最上位ビットプレーンに設定される（処理ブロック1402）。ビットプレーン変数Sを最上位ビットプレーンに設定した後、処理ロジックは随意にエントロピー符号化器を初期化する（処理ブロック1403）。

【0207】エントロピー符号化器が初期化された後、処理ロジックは各係数の初期値をゼロにする（処理ブロック1404）。次に、処理ロジックは各係数の各ビットをコンテキストモデルでモデリングして復号し（処理ブロック1405）、そして係数をフィルタ処理に適し

た形式に変換する（処理ブロック1406）。この変換は、例えばビット有意度形式から2の補数形式への変換である。続いて、処理ロジックはまず最高の分割レベルからの係数にインバースフィルタをかける（処理ブロック1407）。

【0208】次に、全てのレベルがインバースフィルタ処理されたか判定する（処理ブロック1408）。まだ全レベルはインバースフィルタ処理されていなければ、処理ロジックは、次の最も高い分割レベルの係数にインバースフィルタをかけ（処理ブロック1409）、処理ブロック1408に戻る。全レベルがインバースフィルタ処理されたならば、処理ブロック1410に進み復元データが格納されるか伝送される。復元データを格納又は伝送した後、まだほかに符号化単位があるか判定する（処理ブロック1411）。まだほかに符号化単位があるときには、処理は処理ブロック1401にループバックし、プロセスが繰り返される。もうほかに符号化単位がなければ、プロセスは終了する。

【0209】図20は本発明によるビットモデリングプロセスの一実施例を示す。図20において、ビットモデリングプロセスは最初に係数変数Cを最初の係数に設定する（処理ブロック1501）。つぎに、 $|c| > 2^s$ の判定が行われる（処理ブロック1502）。この判定条件が成立するときには、処理ブロック1503に処理が進み、処理ロジックはテーブルビット用モデルを用いて係数CのビットSを符号化し、処理ブロック1508に処理が進む。このテーブルビット用モデルは静的（非適応的）モデルでかまわない。 $|c|$ が 2^s より大きくなっているときには、処理は処理ブロック1504に進み、処理ブロックはテンプレートをヘッドビット（頭の0のビットと最初の”1”ビット）に適用する。テンプレートを適用した後、処理ロジックは係数CのビットSを符号化する（処理ブロック1505）。可能なテンプレートを図33（A）～（C）に示す。テンプレートは、図24及び図25に示すようにLUTを用いて実現してよい。

【0210】次に、係数CのビットSがオンであるか判定する（処理ブロック1506）。係数CのビットSがオンでなければ、処理ブロック1508へ進む。一方、係数CのビットSがオンならば、処理ブロック1507に進み、処理ロジックは符号(sign)ビットを符号化する。それから処理ブロック1508へ進む。

【0211】処理ブロック1508で、係数Cが最後の係数であるか判定する。係数Cが最後の係数でなければ、処理ブロック1509に進んで係数変数Cが次の係数に設定され、処理ブロック1502から処理を続ける。一方、係数Cが最後の係数ならば、処理ブロック1510に進んでSが最後のビットプレーンであるか判定される。Sが最後のビットプレーンでなければ、ビットプレーン変数Sが1だけデクリメントされ（処理ブロック1511）、処理ブロック1501から処理が継続す

る。Sが最後のビットプレーンとときは、処理は終了する。

【0212】＜他の実施例＞本発明は、ハードウエア及び/又はソフトウエアで実現できる。ハードウエアにより本発明を実施するには、ウエーブレットフィルタ、同フィルタにデータを供給するためのメモリ/データフロー管理、本発明の埋め込み符号化を制御するためのコンテキストモデル、同コンテキストモデルにデータを提供するためのメモリ/データフロー管理、及び2元エントロピー符号化器を実現する必要がある。

【0213】＜ウエーブレットフィルタ＞本発明のフォワード・ウエーブレットフィルタの一実施例が図21に示されている。図21に示したウエーブレットフィルタは、 $x(2) \sim x(5)$ として示した16ビットの2の補数の入力画素を4個取り込む。

【0214】図21において、2タップ”11”低域通過フィルタは16ビット加算器1601を用いる。その出力はそれぞれSとDと呼ばれる。この加算器の出力（S）は、1ビットシフトブロック1603により16ビットに丸められる。この1ビットシフトブロック1603は、17ビット入力を1ビット右シフトすることにより2による割り算を実行する。

【0215】6タップ”-1 -1 8 -8 1 1”高域通過フィルタは、 $-S_0 + 4D_1 + S_2$ の計算を必要とする。 $S_2 - S_0$ は、1ビットシフトブロック1603の出力及び $Y_0(0)$ を受け取る16ビット減算器1605により計算される。4D1項は、減算器1602、2ビットシフトブロック1604及び加算器1608によって計算される。16ビット減算器1602の出力は、2ビットシフトブロック1604で2ビット左シフトされることにより4倍される。2ビットシフトブロック1604の出力は、加算器1608により2と加算される。なお、2ビットシフトがあるため、加算器1608は布線に置き換えられてもよい。加算器1608の4D1出力と減算器1605の出力の加算が20ビット加算器1606により行なわれる。この加算器1606の出力は、2ビットシフトブロック1607によって18ビットに丸められる。この2ビットシフトブロック1607は、その20ビット入力を2ビット右シフトすることにより4で割る働きをする。

【0216】このように、必要とされる演算ハードウエア全体（中間結果を格納するためのレジスタは数にいれない）は、

- ・16ビット加算器 1個
- ・16ビット減算器 2個
- ・19ビット加算器 1個

である。なお、シフト操作（及び加算器1608）は布線によって行なわれる所以ロジックは全く必要とされない。

【0217】他の実施例では、Nビットサイズの入力の

場合、1個のNビット加算器、2個のNビット減算器及び1個の(N+3)ビット加算器が使われることになる。

【0218】これらの加算器／減算器はハードウエアコストが非常に低いので、希望するならばフィルタの並列構成も利用できる。

【0219】あるいは、 $x(3)$ と $x(2)$ の減算の代わりに、 $x(4) - x(5)$ を計算し、これを次のシフトまたはフィルタ処理のための $x(2) - x(3)$ として必要になるまでセーブしてもよい。このフォワードフィルタ及び下に述べるインバースフィルタのいずれも、スループットを上げるためパイプライン化してもよい。

【0220】インバース・ウェーブレットフィルタが図22に示されている。入力 $Y_0(0)$ 、 $Y_0(2)$ は減算器1701により減算される。減算器1701の出力に、加算器1709により2が加算される。この加算の結果は2ビットシフトブロック1702で右に2ビットシフトされる。これは実質的には、減算器出力を4で割り算することである。2ビットシフトブロック1702の出力と入力 $Y_1(0)$ との減算が減算器1704で行なわれる。入力 $Y_0(1)$ は、1ビットシフトブロック1703により左に1ビットシフトされることにより、2倍される。 $Y_0(1)$ が1ビットシフトされた(2倍された)のち、そのシフト後の値のLSBは減算器1704の出力から差し引かれるLSBであり、1ビットシフトブロック1703の16ビット出力と結合されて加算器1705と減算器1706の入力となる。加算器1705及び減算器1706のもう一方の入力は、減算器1704の出力である。加算器1705及び減算器1706の出力はその後、クリップユニット1707、1708によりクリッピングを施される。

【0221】2つのクリップ操作は様々なものを用いてよい。いずれの場合も、20ビットの値が1ビットシフトされ(2で割り算され)、19ビット値とされる。非損失性圧縮だけをするシステムの場合、下位16ビットを出力してよい(残り3ビットは無視してよい)。損失性システム(または損失性／非損失性システム)では、19ビット値は、負ならば0に設定され、 $2^{16}-1$ を超えるときには $2^{16}-1$ に設定され、それ以外ならば下位16ビットを出力してよい。

【0222】Nビットサイズの入力の場合、1個のNビット減算器、1個の(N+1)ビット加算器、1個の(N+2)ビット減算器、1個の(N+3)ビット加算器及び1個の(N+3)ビット減算器が使われることになる。そして、クリップユニットはNビットを出力する。

【0223】ウェーブレット変換の一実施例では、変換計算にモンテカルロ除法が利用され、擬似乱数発生器を用い、その出力に基づき変換演算の結果が切り上げ又は切り下げる。丸めが行われたことを復号器が知つて

いる限り(すなわち、復号器が同一点から始まる同一の乱数発生器を用いる限り)、そのような構成が利用できる。

【0224】<ウェーブレットフィルタのデータフロー管理のためのメモリ使用量>本発明のウェーブレットフィルタのためのメモリ及びデータフロー管理に関してであるが、1枚のフルフレームがメモリにぴったり入る画像の場合、メモリ／データフロー管理は難しい問題ではない。 1024×1024 、16ビットの医用画像(すなわち2Mバイトサイズ)の場合でもそうであるが、1つのフルフレームバッファを用意すれば多くの用途に間に合う。さらに大きな画像(例えば、A4、400DPI、4色の画像は約50Mバイトの大きさである)については、限られた量のラインバッファメモリを利用してウェーブレット変換を行なうのが望ましい。

【0225】なお、本発明が1パスシステムを実現するのにフルフレームバッファは必要でない。このため、必要とされるメモリは約100分の1になる(大きな画像のためのフルフレームバッファを用いる場合に比べ)。本発明の1パスシステムは後述する。

【0226】フィルタメモリに格納されるデータは、埋め込み符号化及び2元エントロピー符号化がなされる係数の系列である。埋め込み符号化は、コンテキストを用いて水平符号化の利用を調整し、かつデータを適当な順序で供給する。コンテキストモデルを用いる。このコンテキストモデルは、メモリ管理スキームと関連して働く。フルフレームメモリを持つシステムの場合、データを適当な順序で供給することは難しくない。

【0227】システムの作業領域メモリの大きさが限られている場合、一実施例では、記憶に必要なメモリの作業領域ライン数を減らすため色々な高さ(height)の変換が使われる。このように、幅の大きな画像がきても、その画像を割り当てられた作業領域メモリの範囲内に効率よく圧縮することができる。例えば、ライン数を減らすためS変換は垂直に用いられる。

【0228】ラスターデータをバッファしウェーブレット変換を実行できるようにするために、メモリが必要である。このメモリを最小にすることが、アプリケーションによってはコスト削減のため重要である。それを実現する方法を下に述べる。

【0229】ここに述べるウェーブレット2-D変換の一実施例は、1-パス(pass)構成でメモリ使用が限定される場合を想定している。一実施例では、ピラミッド分割を実行するために適用されるウェーブレット変換は、TS変換とS変換だけである。この実施例では、4レベルの分離可能なピラミッド分割がある。一実施例では、4レベル分割がS変換及びTS変換によって実行される。一実施例においては、水平(行方向)分割にはTS変換だけが使われる、すなわち水平分割はTS-TS-TS-TSの形になる。垂直(列方向)分割には、S変

換とTS変換の両方が、TS-TS-S-Sの形で使われる。その2つのTS変換がS変換で置き換えられると、圧縮率の犠牲は小さいがメモリ使用に大きな影響がある。水平変換と垂直変換は、例のごとく交互に適用される(図26)。

【0230】S変換とTS変換をどのように組み合わせて水平変換及び垂直変換を実施してもよい。また、変換の順序が色々に変わっても構わないが、復号器が、その順序を認識し逆の順序で逆の動作をしないと完全には可逆とならない。

【0231】<係数ツリー>ピラミッドシステムでは、ツリー構造を用いて係数を複数組にグループ分けできる。各ツリーのルートは純粹に1つの低域通過係数である。図10は、変換画像の1つの純粹に低域通過係数に関するツリー構造を示している。画像のような2次元信号の場合、ツリーのルートは3個の“子”を持ち、その他のノードはそれぞれ4個の子を持つ。ツリー階層構造は2次元信号に限定されない。例えば、1次元信号の場合、ルートは1個の子を持ち、ルート以外のノードはそれぞれ2個の子を持つ。1次元及び2次元のケースから、より多次元のケースを理解できる。

【0232】ツリー構造は、図11乃至図13に示したフィルタの動作からも明らかである。フィルタ・ペア群の作用とサブサンプリングにより、前述の係数が関連付けられる。

【0233】一実施例では、係数はビット有意度又はビットプレーン埋め込み方式で符号化される。係数は最上位のビットプレーンから最下位のビットプレーンへと符号化されるので、データのビットプレーン数を確定しなければならない。本発明では、係数値の絶対値の上限を、データから計算するか、画像の深度(depth)及びフィルタ係数より得ることによって、ビットプレーン数の確定がなされる。一例を挙げれば、上限が149のときは、8ビットの有意度つまり8つのビットプレーンがある。ソフトウェアの速度のために、ビットプレーン符号化を利用できないかもしれない。別の実施例では、係数が2進数として意味があるときにだけ、ビットプレーンが符号化される。

【0234】一実施例では、本発明の水平コンテキストモデルは、ウェーブレット係数をビット有意度埋め込み符号化して2元エントロピー符号化器に送る。

【0235】<コンテキストモデルの代案>分割が終わりデータ係数が順序付けされたならば、本発明のコンテキストモデルが係数を符号化するために用いられる。様々なコンテキストモデルを用いることができる。デシジョンは、空間的位置、レベル、及び/又はビット位置により条件付けすることができる。デシジョンはまた、現在データと空間的位置、レベル及び/またはビット位置が接近した、既に符号化されたデータにより条件付けするこどできる。

【0236】いくつか例を挙げれば次のとおりである。最上位のテールビット(したがって最も容易に予測される)は、その他のビットとは異なったコンテキストを用いることができる。ヘッドビットは、同一変換レベルの空間的に接近した前の係数の同一ビットにより条件付けすることができる。同様に、有意な係数の符号ビットは、同一レベルの空間的に接近した前の係数の符号ビット、あるいは親の係数の符号ビットにより条件付けすることができる。

【0237】コンテキストモデルの改良は、空間的構造またはマルチ解像度構造を有する画像を圧縮する場合に特に重要である。線図形またはテキストのグレースケール画像は、その両方の構造タイプを持つ画像の一例である。コンテキストモデルの改良は、すでに圧縮されてから伸長されたファイルを、指定したピーク誤差で圧縮する場合にも重要である。

【0238】本発明をソフトウェアで実施する場合、コンテキストのためのビットを得るために長い時間が費やされるが、それは、そのビットが条件付けに欠かせないからである(例えば、すべてのヘッドビット)。本発明の一実施例によれば、ルックアップテーブル(LUT)を使うことによりソフトウェアによる実行をスピードアップし得る。これは、コンテキストとして利用される北(N)画素、北西(NW)画素、西(W)画素及び南西(SW)画素に対し別々にビット抽出操作をしないようとする。

【0239】図24及び図25は本発明におけるヘッドビットの条件付けのための状態マシンを示している。図24において、新しい親のためのLUT1901が符号化/復号ブロック1902に接続されている。LUT1901は、親を表すビットと、上の(NE)係数、現在の(E)係数及び下の(S)係数を表すビットを受け取るように接続される。一実施例では、親入力と現在入力はそれぞれ2ビットからなる。LUT1901の他の入力に、LUT1901より出力されるコンテキストと符号化/復号ブロック1902の出力の全部又は一部がフィードバックとして入る。一実施例では、LUT1901によるコンテキストとしての10ビット出力の8ビットがLUT1901の入力にフィードバックされる。

【0240】NE, E, S係数が用いられるのは、それら係数が、前ビットプレーンに関連した係数情報からなるテンプレート情報の前エッジを表すからである。なお、南(S)係数に代え、南東(SE)係数を用いてよい。

【0241】一実施例では、テンプレートが符号化単位の外側にあるときには、その外側の条件付けビットを、現在画素のビットで置き換えてよい。

【0242】図25は、同じ親に対しLUTを用いて条件付けする状態マシンを示している。その場合、コンテキスト全体がLUT1903の入力へフィードバックさ

れる。1904は符号化／復号ブロックである。

【0243】データがラスター順に処理される場合、LUTを利用するとメモリアクセス数が減るが、それは最後のコンテキストを生成するため使われた同じメモリを再格納しなくてよいからである。

【0244】また、LUTメモリのサイズを減らすため、親の条件付けを、他の条件付けだけを扱うLUTの出力と論理和することにより独立に行つてもよい。

【0245】少し大きなLUTテーブルにより、次のビットプレーンに対する条件付けの大部分も可能になる。別の小さなLUTで、現在コンテキストLUTから状態情報を取得し、それを次のビットプレーンから新たに入手できるデータと結合することも可能である。これは、1つのツリーを一度に符号化する場合に特に効果的である。

【0246】本発明に関し以上に述べたように、"効率的"とは、変換の行列式の値が1であることを意味すると定義してよい。かかる場合、低い確率の事象のためのスペースを、その低確率がゼロの時に省くことにより、符号空間は無駄に使われない。しかし、同じく8ビットの係数が入力し、1つの8ビット係数と1つの9ビット係数を発生するので、依然として効率は向上するであろう。あらたな非能率な点は、可能な係数の空間が回転することによる。

【0247】変換操作の一定の結果から、計算に使われた数が一意に明らかになることに注意すべきである。そのようになるのは、それら結果が可能な結果のレンジの境界に近い時である。これを図23に例示する。図23において、uは低域通過値を表し、vは高域通過値である。uとvの値は独立でないので、これらの値は統合情報を考慮するとエントロピー符号化が容易になろう。その理由は、図23に示されるように、殆どの低域通過値について、高域通過値用符号空間のかなりが使われないからである。多くの用途では、これら可能なペアに割り当てる確率が低いかからといって利益は殆どない。しかし、用途によっては、それなりの利益があるかもしれない。操作を高速化するために、LL係数のもっと多くのビットをLH係数、HL係数及びHH係数よりも前に送つてもよい。こうすると境界設定が容易になる。

【0248】実施例によっては、各符号化単位の符号化が済むと、すべてがリセットされ、次の単位を符号化する時に全ての統計量と確率がリセットされる。一実施例では、統計量の一部又は全部がセーブされる。これらの統計量は、その後の符号化単位が始まる時に初期統計量として働く。一実施例では、最初の又は前の符号化単位の符号化中の所定時点に統計量がセーブされる。例えば、第3のビットプレーンを符号化した後、現在符号化単位の符号化に使われた統計量がセーブされ、次の符号化単位又はそれより後の符号化単位の符号化の開始時に統計量として用いられる。別の実施例では、全ての画像

の種類が評価され、ハードにより符号化された統計量の集合が決定される。そして、これらハード符号化統計量をデフォルトとして用いて符号化が実行される。もう一つの実施例では、統計量がビットプレーン毎にセーブされ、その統計量が他のタイル(tile)の類似したビットプレーンの符号化時に利用される。

【0249】一実施例では、最初の1ビットまでは全く符号化がない。係数の最初の1のビットが現れるた時に、符号(sign)が符号化される。ヘッドビットは画像／領域に依存するが、テールビットは様々な画像及び領域にわたって、それより一定している。テールビットが(ヘッドビット中の)初めの1のビットから、どのくらい離れているかに基づいて、ある確率クラスがテールのビットを符号化するために用いられる。一実施例では、係数の最初のテールビットは0.7を含む確率クラスで符号化される。2番目及び3番目のテールビットは0.6を含む確率クラスで符号化される。最後に、4番目以降のテールビットは0.5を含む確率クラスで符号化される。

【0250】<A. ウエーブレット変換の実行>1-パスのシステムでは、実行されるウエーブレット変換は圧縮性能とメモリ使用量との兼ね合いで決まる。符号化単位のサイズは、ラインバッファが最小でメモリ使用量が最小となるように選ばれる(画像がラスター順に与えられるとして)。ウエーブレット変換の中間係数は、同じメモリに適当な時に入力と置き換わって格納される。

【0251】<ウエーブレット変換フィルタの選択>ここに述べるウエーブレット2-D変換は、1-パスで実行されかつメモリ使用量が制約されることを予定している。4レベルの分離可能なピラミッド分割がある。水平分割には、TS変換だけが用いられる、すなわち、水平分割はTS-TS-TS-TSの形となる。垂直分割には、S変換とTS変換が両方用いられ、そして垂直分割はTS-TS-S-Sの形となる。水平変換と垂直変換は、例の通り交互に適用される。図26は水平分割と垂直分割を表している。

【0252】そのTS変換の2つがS変換で置き換えられると、圧縮率の犠牲は小さいがメモリ使用量に大きな影響がある。最後の2つの垂直パスにS変換を使うことを選んだのは、単に使用メモリを減らすためである。S変換を使うと、係数バッファの約32ラインが節約される(例えば、48ラインから16ラインに減る)。なお、全ての分割にTS変換を使えば、より優れた圧縮性能を得られる。

【0253】<符号化単位の定義>一実施例では、符号化単位は1列のツリー(1列のLL係数とそれらの子孫全部)により定義される。4レベルの分割の場合、それは空間領域において符号化単位は16ライン×画像幅であるということを暗に意味する。図27は1つの符号化単位を示している。なお、図27は縮小率が一定ではな

い。レベル1のブロックは1回の2-D分割の後の画像である。何度も繰り返すのでLL (low-low), LH (low-high), HL (high-low), HH (high-high)という名称が、サブブロックを呼ぶのに用いられ、またレベル1～レベル4のブロック全部に適用される。レベル2のブロックはレベル1のブロックのサブブロックLLを2-D分割した結果である。同様に、ブロック3とブロック4はそれぞれ、レベル2のブロックとレベル3のブロックのサブブロックLLを2-D分割したものである。

【0254】符号化単位は、レベル1のHH, HL, LHの係数については8ラインの高さであり、レベル2の同係数については4ラインの高さであり、レベル3の同係数については2ラインの高さであり、レベル4の同係数及びサブブロックLLの係数については1ラインの高さである。なお、ステップ毎に解像度が下がるにつれて、長さ並びに行数が半分になる。レベル4のブロックのLL内の各係数は、1つのツリーの一番上の親である。

【0255】<バッファリング及び係数計算>図27で述べた1つの符号化単位を生成するためには、 $2 \cdot w \cdot m$ の大きさの作業領域バッファが用いられる。ここで、wは画像の幅、mは最大係数サイズ（ビット数）である。垂直変換（つまり列方向）のパスのために選ばれるウエーブレットフィルタの性質から、必要な作業領域メモリは約18～20ラインである。各水平変換（つまり行方向）パスは、すべてTS変換であるが、1ライン（行）ずつ計算され、新たな係数が古い係数もしくは画素に取って代わる。

【0256】初めの2つの垂直変換パスはTS変換フィルタを使う。6タップの高域通過フィルタであるから、垂直パスの各高域通過係数は6ラインの画素又は係数データによって決まる。生成された高域通過TS係数は、オーバーラップのため最も上の2ラインと下の4ラインに関係する。これが図28に示されている。図28には、1つの符号化単位の1つの垂直画像セグメントが示されている。この垂直画像セグメントは、オリジナル画像をTS変換の1回の水平パスにより変換した結果である。符号化単位の垂直レベル1セグメントが示されているが、これは画像の第1レベルの2-D変換である。この垂直レベル1セグメントは、TS変換で水平パスを実行することにより得られる。符号化単位の垂直レベル2セグメントも示されているが、これはレベル1ブロックのLLサブブロックに対してTS変換を両次元につき適用して得られたものである。

【0257】TS変換は4画素（又は4係数）だけオーバーラップさせられるので、ある符号化単位の終わりで4ライン分のデータが、次の符号化単位の係数計算に利用すべくセーブされることに気づかれない。つまり、レベル1の係数を生成するために、最上部と最下部の両方で割り増しの2ライン分の画素が必要とされ、あるいは

は、最下部で割り増しの4ライン分の画素が必要とされる。レベル2の係数を生成するためには、割り増しの2ライン分のレベル1係数が最上部と最下部の両方で必要とされ、あるいは割り増しの4ライン分のレベル1係数が必要とされる。これら割り増しのレベル1係数を生成するために、別の2ライン分の画素が最上部と最下部の両方で必要とされ、または別の4ライン分の画素が最下部で必要とされる。したがって、各符号化単位は28垂直ラインにわたる。

【0258】しかし、重要なことは、これらの”割り増し”のレベル1係数を生成するために余分な計算が必要とされないことである。というのは、それら割り増しのレベル1係数は、現在の符号化単位の上又は下の符号化単位で利用されることになるからである。また、レベル2係数だけが格納されるので、20ライン分のメモリしか必要とされないことにも気づかれない。

【0259】最後の2つの垂直パスはS変換であり、これは低域通過のオーバーラップがないので割り増しラインを必要としない。

【0260】<変換計算用メモリ>画素値又は計数値のサイズをbビット（ $-2^{b-1}, \dots, 0, \dots, 2^{b-1}-1$ のレンジ）とすると、S変換とTS変換の平滑出力S（.）もbビットである。つまり、それらの平滑出力は入力と同じレンジを持つ。しかし、S変換とTS変換の1次元細密出力d（.）は表現にb+1ビットとb+2ビットをそれぞれ必要とする。

【0261】図29は符号化単位とともに必要なラインバッファリングの一部を示す。影を付けた領域と矢線は、現在符号化単位の一部であって現在の符号化のためにメモリにセーブされる必要のある係数である。点線の矢線は、符号化単位の係数を計算するのに必要な一時的な係数である。これらの一時的係数は、新しい係数により上書きされる。実線の矢線は、現在符号化単位の計算の副産物たる係数であり、次の符号化単位の一部となるべくセーブされる。

【0262】最終レベル（レベル4）の係数は、4つのサブブロック（LL, LH, HL, HH）全てでたったの1ラインである。垂直変換だけであるが、レベル3からレベル4を計算するためにS変換が用いられるので、全てのサブブロックはレベル3の係数を2ライン分しか必要としない。同様に、レベル2からレベル3を計算するには、4ライン分のレベル2係数を必要とする。これらの係数は全て現在符号化単位の一部である。

【0263】レベル2とレベル1の垂直パスの計算にTS変換が用いられる。6タップ高域通過オーバーラップフィルタのオーバーラップ特性のため、これらレベルは次の符号化単位からのデータを必要とする。このデータは、現在符号化単位の係数の計算に用いられ、そして次の符号化単位で利用するためセーブされる。

【0264】レベル1からレベル2の高域通過サブブロ

47

ック (LH, HH) を計算するためには、12ラインが必要である (ダウンサンプリングにより 4ラインになる 8ラインとオーバーラップのための割り増しの 4ライン)。これらのラインは、図 29 のレベル 1 の低域通過サブブロック (LL, HL) に、現在符号化単位の一部である 8ライン、及び、次の符号化単位の一部である 4ラインとして示されている。

【0265】レベル 1 の低域通過サブブロック (LL, HL) の 12ラインを計算するためには、0 レベルからの 24ラインが必要とされる。これらのレベル 0 の 24ラインは、レベル 1 の高域通過サブブロックの 10ラインを生成することができる (ダウンサンプリングにより 8ラインになる 16ラインとオーバーラップのための割り増しの 4ライン)。これらラインの 10ライン全部を、現在符号化単位にとつては 8ラインしか必要でないが、レベル 1 で計算し格納するのが最も効率的である。

48

よって、オーバーラップ用の割り増し 4ラインだけをレベル 0 でセーブする必要がある。

【0266】画素深度(depth)が b の画像から始めるとして、分離可能な 2-D 変換で、行変換及び列変換がともに TS 変換の場合、LL, HL, LH, HH サブブロックの係数のレンジ (値域) はそれぞれ b, b+2, b+4 である (図 16)。この分離可能な 2-D 変換が水平 TS 変換と垂直 S 変換からなる場合、LL, HL, LH, HH の係数のレンジはそれぞれ b, b+1, b+2, b+3 である。表 2, 表 3, 表 4 及び表 5 は各ブロックに必要とされるメモリの計算を示している。なお、各ブロック毎に幅 w の画像が 1 個として、ビット単位でサイズを計算したものである。

【0267】

【表 2】

レベル 0 のためのメモリコスト

サブブロック	L	H
メモリ	$4 \cdot b \cdot w / 2$	$4 \cdot (b+2) \cdot w / 2$

【0268】

【表 3】

レベル 1 のためのメモリコスト

サブブロック	LL	HL	LH	HH
メモリ	$4 \cdot b \cdot w / 2$	$10 \cdot (b+2) \cdot w / 2$	$12 \cdot (b+2) \cdot w / 2$	$10 \cdot (b+4) \cdot w / 2$

【0269】

【表 4】

レベル 2 のためのメモリコスト

サブブロック	LL	HL	LH	HH
メモリ	$0 \cdot b \cdot w / 4$	$4 \cdot (b+2) \cdot w / 4$	$4 \cdot (b+2) \cdot w / 4$	$4 \cdot (b+4) \cdot w / 4$

【0270】

【表 5】

レベル 3 のためのメモリコスト

サブブロック	LL	HL	LH	HH
メモリ	$0 \cdot b \cdot w / 8$	$2 \cdot (b+1) \cdot w / 8$	$2 \cdot (b+2) \cdot w / 8$	$2 \cdot (b+3) \cdot w / 8$

【0271】

【表 6】

レベル 4 のためのメモリコスト

サブブロック	LL	HL	LH	HH
メモリ	$1 \cdot b \cdot w / 16$	$1 \cdot (b+1) \cdot w / 16$	$1 \cdot (b+2) \cdot w / 16$	$1 \cdot (b+3) \cdot w / 16$

【0272】上記数を全部足すと $(26b + 557/8) \cdot w$ ビットとなり、丸めれば $(26b + 56) \cdot w$ ビットである。最大サイズ b+4 ビットの 2 ラインの計算バッファの分である。2 \cdot (b+4) を加えると、合計メモリコストは $(28b + 64) \cdot w$ ビットになる。例えば、8 ビット、512 画素幅の画像の場合、147, 456 ビットつまり約 18 K バイトのメモリが必要となる。

【0273】一実施例では、変換のサイズは、画像の幅と利用可能なメモリの固定サイズに基づいて選ばれる。つまり、あるサイズの画像が本発明のシステムに入力されると、利用できる変換メモリの大きさが限られているために、分割レベル数が決まってしまう。より大きなメモリを利用できれば、分割レベル数が増加する。なお、このことは、画像がシステムに受け取られている間に動的に起こるかもしれない。十分なメモリを利用可能なら

ば、追加レベルの分割を行うためにLL係数がウェーブレットフィルタを通じて送り込まれる。なお、レベル数を増減させることによる影響の一つは、達成される圧縮量を増減させることである。

【0274】<B. 本発明の埋め込み順序>

<符号化ストリームの順序付け>図30は、符号化ストリームの順序付けと符号化単位内における順序付けを示している。図30において、ヘッダー(header)2401の後に、符号化単位2402が上から下へ順に続く。1つの符号化単位内において、LL係数2403は符号化されておらずラスター(ライン)順である。LL係数の後に、1ビットプレーンずつエントロピー符号化したデータが、最上位のビットプレーンから最下位のビットプレーンへの順に続く。この時、すべての係数からの第1ビットプレーンが符号化され、次に第2ビットプレーンが符号化され、以下同様である。

【0275】<係数のアラインメント(alignment)>本発明の一実施例では、コンテキストモデルは正規化されていない $1+Z^{-1}$ 低域通過フィルタを用いる。しかし、このコンテキストモデルを正規化フィルタ、例えば(数33)のフィルタと一緒に用いてもよい。

【0276】

【数33】

$$\frac{1+Z^{-1}}{\sqrt{2}}$$

【0277】正規化フィルタを用いるために、フォワードフィルタ1600とコンテキストモデル105の間のアラインメント(alignment)ユニットを用いて、非正規化フィルタより獲得した(あるいは失った)エネルギーを補償してもよく、これは圧縮を向上させる。アラインメントは損失正動作のための非一様量子化を許容するので、アラインメントは損失性復元画像の視覚的品質を向上させることができる。1次元の場合、ツリーの各レベルからの係数は様々なアラインメントを持つことになる(除数= $\sqrt{2}$, 2, $2\sqrt{2}$, 4, 乗数= $2\sqrt{2}$, 2, $\sqrt{2}$, 1)。2次元の場合、除数は2, 4, 8, 16、

係数アラインメント

1-HH	1-HL,1-LH	2-HH	2-HL,2-LH	3-HH	3-HL,3-LH	4-HH	4-HL,4-LH
基準	左1	左1	左2	左2	左3	左3	左4

【0282】様々なサイズの係数に関するアラインメントは、符号化器と復号器の両方に知らされるので、エントロピー符号化器の効率には影響を与えない。

【0283】また、符号化単位の全てのブロックの全てのサブブロックは、それ固有の最大の絶対値レンジを持ち、そのレンジを符号化器及び複合化器が知っていることに注意されたい。殆どのサブブロックに、いくつかの確定的な2進ゼロ値があり、それらは効率を上げるためにエントロピー符号化器でスキップされる。

乗数は8, 4, 2, 1となろう。

【0278】アラインメントは単に同様の2元デシジョンを符号化するためにグルーピングしようとするものであるので、厳密な正規化値を用いることは重要でない。復号期間にはアラインメントを逆にしなければならないので、乗算と除算の両方が必要になる。2のべき乗の因数/除数を用いると、ハードウェアによる効率的なシフト操作が可能になろう。係数に2のべき乗が掛け合わされた時には、有意度の小さい、追加された0ビットは符号化する必要がない。

【0279】チューニング及び精密で非一様な量子化のために係数アラインメントを利用することができる。画像(2次元信号)の場合、RTS変換の一実施例は、図17に示した数を周波数帯域に乗じることにより係数を整列させる。これらの数を乗じることによって、RTS変換は、TS変換の正確な復元ウェーブレットの極めて近い近似値となる。

【0280】この1-パスの実施例は、様々なフィルタ組合せに対しMSEの点で最適な1つのアラインメントだけを用いる。表7はアラインメント数を示す。係数はビット有意度別に符号化される。ここで、第1のビットプレーンは全ての係数の最も左側の絶対値ビットである。各係数の符号(sign)ビットは、当該係数が非ゼロの絶対値ビットを持つ最も上位のビットプレーンまでは符号化されない。つまり、符号ビットは、最初の”オン”ビットが符号化された直後に符号化される。こうすることの利点は、ゼロ絶対値を持つ係数の符号ビットを符号化しないこと、そして、符号ビットを、埋め込み符号ストリーム中の当該符号ビットが関係する点まで符号化しないことである。画素深度がbの画像の場合、可能な係数値の最大値は $2^{b+3}-1$ である、すなわちb+3ビット数である。したがって、係数はすべて、b+3ビットの2元デシジョンに、必要なら符号用の1ビットを加えたものに符号化される。

【0281】

【表7】

【0284】各ビットプレーン期間に係数が処理される順序は、低解像度から高解像度へ、かつ低周波数から高周波数への順である。各ビットプレーン内の係数の順序は、高レベル(低解像度、低周波数)から低レベル(高分解能、高周波数)への順、つまり4 LL, 4 HL, 4 LH, 4 HH, 3 HL, 3 LH, 3 HH, 2 HL, 2 LH, 2 HH, 1 HL, 1 LH, 1 HHの順序である。各サブブロック内においては、符号化はラスタースキャン順である。

【0285】なお、同じデータセットの符号化単位が様々なアラインメントを持つてもよい。一実施例では、アラインメントは、図30のヘッダー2401のようなヘッダーで指定される。

【0286】<水平コンテキストモデル>図32は、符号化単位の全ての係数の隣接係数を表す。図32において、隣接係数は分かりやすい地理的表記法で表されている（例えば、N=北、NE=北東、等々）。

【0287】ある係数、例えば図32のPと、ある現在ビットプレーンが与えられたとすると、コンテキストモデルは所与のビットプレーンより前の符号化単位全てから与えられるどのような情報も利用することができる。本コンテキストモデルの場合、注目係数の親係数も利用される。

【0288】注目係数の注目ビットに対するコンテキストを決定するために隣接係数や親係数の値を利用するとよりむしろ、その情報は本明細書においてテール(tail)情報と呼ぶ2ビットにまとめられる。この情報はメモリに格納しても、あるいは親係数の隣接係数から動的に計算してもよい。テール情報は、最初の非ゼロの絶対値ビットがすでに見つかったか否か（例えば最初の”オン”ビットがすでに見つかったか否か）を示す、そして、すでに見つかっているならば、幾つ前のビットプレーンであったかを示す。表8はテール情報ビットを説明している。

【0289】

【表8】

テール情報の定義

テール	定義
0	オンビットは未検出
1	最初のオンビットは直前のビットプレーンにあった
2	最初のオンビットは2又は3ビットプレーン前にあった
3	最初のオンビットは4ビットプレーン以前にあった

【0290】2ビットのテール情報の”テール・オン”ビットの情報は、そのテール情報がゼロであるか否かを示す。一実施例では、テール情報とテール・オンビットは係数が符号化された直後に更新される。別の実施例では、その更新は、並列コンテキスト生成を可能にするため、もっと遅れて行われる。

【0291】一例として、表9は、ある係数に対するテール・オンビットを、ビットプレーンの関数として示しているが、絶対値は

【0292】

【外8】

0	0	0	1	*	*	*	*
---	---	---	---	---	---	---	---

【0293】のように2進法で表現される（”*”は1でも0でもよいことを意味する）。

【0294】

【表9】

コンテキスト係数例に対するテール情報								
ビットプレーン	1	2	3	4	5	6	7	8
係数例が出現する前	0	0	0	1	2	2	3	3
係数例が出現した後	0	0	0	0	1	2	2	3

【0295】3つ目のコンテキスト情報は符号(sign)ビットである。符号ビットは最初のオン・ビットのすぐ後に符号化されるので、テールは符号情報が既知か否かを示す。したがって、テールが非ゼロでなければ符号ビットは情報コンテキストを持たない（符号(sign)には3つの可能性がある、つまり正か、負か、あるいは未知かであることを思い出されたい）。

【0296】システムのコンテキストモデルは、最高1ビットを使ってコンテキストを記述する。この1ビット数が全部指定されるわけではない。すなわち、実際

的には、符号(sign)ビットコンテキストを含め1030個又は1031個のコンテキストしか利用されない。どのビット位置の意味も前の2進値に左右される。一実施例は次のルールに従う。

【0297】注目係数のテール・オンビットが0ならば（ヘッドビットの場合）、親及びW係数のテール情報ビットとNW, N, NE, E, SW, S係数のテール・オンビットからの1024コンテキスト。一実施例では、ヘッドビットに対し適応符号化が用いられる。実施例によつては、ただ1つのコンテキストを用いてヘッドビッ

トの何らかの”ラン(run)符号化”を提供する。符号化すべき次の16ビットは全てヘッドビットで、それらのN, S, E, W隣接係数及び親係数が全てテール情報0を持つならば、ただ1つのデシジョン(decision)が符号化される。このデシジョンは、その符号化すべき16ビットのどれかが現在ビットプレーンに1のビットを持つか否かを示す。1のビットがなければ、その普通に符号化された16デシジョンをスキップすることができる。次の16係数のどれかが、その最初の有意なビットを含んでいるならば、16デシジョンがビット毎に1個ずつ用いられる。この”ルックアヘッド”(look ahead)により2元エントロピー符号化器に対するコールが減り、結果として高速化するとともに圧縮率が向上する。

【0298】注目係数のテール・オンビットが1のときには(テールビットの場合)、テール情報ビットからの3コンテキストが係数を与える。前述のように、固定確率符号化を用いてよい。

【0299】注目係数の注目ビットが最初の非ゼロの絶対値ビットであれば、注目係数の符号(sign)ビットが直後に符号化される。符号ビットに対するコンテキストは、N_テール・オンビット及びN_符号ビットからの3コンテキストであり、N_テール・オンビットが0ならば、N_符号ビットは未知である。N_符号ビットが未知ならば、符号(sign)は確率0.5で符号化される。未知でなければ、符号は適応的に符号化される。

【0300】要するに、同じ符号化単位における現在係数、隣接係数及び親係数から得られる情報に基づき、コンテキストを表す11ビット数が生成される。

【0301】図33(A)～(D)は、係数Pの条件付けに利用される、原因(causal)係数及び非原因(non-causal)係数を示す。図示のテンプレートはそれぞれ、テール・オンビットとテール・オン情報を両方利用する。各係数のテール・オンビットは1ビットであるが、各係数のテール・オン情報は2ビットからなる。図33(A)において、テンプレートにより提供されるビット総数は8である。図33(B)及び(C)において、テンプレートにより提供されるビット総数は10である。

【0302】係数Pのヘッドビットを条件付けするためには、追加のビットを用いてよい。一実施例では、2ビットの追加ビットでビット位置を次のように指定する。

0 0	最初のビット(MSB)及び2番目のビット
0 1	3番目のビット及び4番目のビット
1 0	5番目のビット及び6番目のビット
1 1	他のビット

【0303】なお、隣接係数及び親係数に基づいて他のテンプレートを作ることもできる。さらに、一実施例では、係数Pの条件付けに利用される係数は、位置、ビットプレーンが違っても、原因係数である。

【0304】一実施例では、S変換の親が条件付けに用

いられ、TS変換の親は用いられない。こうすると、低域通過ラインを次の低域通過ラインの符号化までセーブすることにより、条件付けのために必要なバッファリングが少なくなる。これは、エントロピー符号化の順序が重要で符号化器のメモリが重要でない場合には有利でない。

【0305】なお、よりスキーの強いデータを生成するため、より多くのコンテキストを持つことと、コンテキスト内のデータを少なくすることによりもたらされる適応効率との間には、トレードオフの関係がある。

【0306】一実施例では、条件付けの必要のないテールビットは(条件付けするために)バッファしなくてよい。それらは利用可能になると即座に符号化できる。そのような場合、通信路マネージャ(manager)は、ただちにそれらビットを通信路へ送出してよい。

【0307】一実施例では、最も低い分割レベルの係数は、他の係数と同じ方法で符号化し、たり、全く符号化しないのではなく、DPCMのような予測符号化により符号化されるであろう。

【0308】テールビットの符号化には、固定確率又は適応的確率のどちらかが用いられるであろう。

【0309】条件付けであるが、最後のビットは、部分的に、その一つ前のビットに基づき条件付けされるであろう。また、最初の”オン”ビットの後のビットは、それらが最初の”オン”ビットからどのくらい離れているかに応じて条件つけられるであろう。

【0310】一実施例では、テールビットの中には適応的に符号化されるものもある。例えば、係数中のテールビットがTビットより少ない時には(例えば、T=2, T=3)、適応符号化が使われる。これらビットのコンテキストは、ビット位置と現在係数中の既に符号化されたテールビットを含んでいる。これは、DPCMデータのためのLangdonの教えるM元中心符号化と同様である。

【0311】別の実施例では、一部または全部のデータが、2元エントロピー符号化器でなくM元エントロピー符号化器により符号化される。M元エントロピー符号化器としては、Tunstall符号化器、固定ハフマン符号化器、適応ハフマン符号化器等がある。例えば、1つのハフマン符号をヘッドビットとして用いることができる。別の実施例では、ヘッドビットを1ビットずつ符号化するのではなく、プライオリティ・エンコーダーを用いて最初の”オン”ビットの位置を調べ、そして、その位置の2進表現のビットが2元エントロピー符号化器によって符号化される。

【0312】<水平コンテキストモデル>本発明のコンテキストモデルがブロック図として図34に示されている。コンテキストモデル2700は、符号(sign)/絶対値ユニット109(図2)と、係数中の別々のビットを処理するための3つのユニットを含む。符号化されるビ

ットに基づき、3ユニット中の1ユニットが選択される。それらユニット間の切り替えをハードウェアで行いやすくするためのスイッチが含まれるかもしれない。これらユニットは、ヘッドビットブロック2701、符号(sign)ビットブロック2702、テールビットブロック2703である。ヘッドビットブロック2701、符号ビットブロック2702及びテールビットブロック2703はそれぞれ、前述のように、ヘッドビット、符号ビット及びテールビットのモデリングする。これら3ユニットの出力はエントロピー符号化器104(図1)へ送られる。

【0313】符号化器は、状態をセーブし(オプション)、初期状態を与え、符号化器を(例えば、1つの符号化単位の終わりで)リセットする、オプションの制御部を含むであろう。

【0314】上に定義したコンテキストは、いくつか例外はあるが、適応2元エントロピー符号化器とともに用いられる。ヘッドビット(注目係数のテール・オンビット=0)とN_テール・オン=1の時の符号ビットに関するコンテキストを採用してよい。

【0315】しかし、テール・オン=1の後のビット及びN_テール・オン=0の時の符号ビットは、定常ソースによってモデリングされる。これらのケースでは、エントロピー符号化器の適応化機能は、必要でないし、実際には圧縮効率を低下させる原因になる可能性がある。以下のコンテキストには、固定(非適応)状態(Qコーダーの状態に関する用語で述べる)が用いられる。

【0316】<統計的モデル>コンテキストは、N_テール・オン=0(N_係数の符号が未知)の時の符号ビットの符号化は固定Qコーダーの状態0-約0.5の確率で符号化されるというものである。

【0317】コンテキストは、最初の非ゼロのビットの後の最初の2進値の符号化(テール情報=1)は固定Qコーダーの状態4-約0.7の確率で符号化されるというものである。

【0318】コンテキストは、最初の非ゼロのビットの後の2番目と3番目の2進値の符号化(テール情報=2)は固定Qコーダーの状態3-約0.6の確率で符号化されるというものである。

【0319】コンテキストは、最初の非ゼロのビットの後の4番目以降の2進値の符号化(テール情報=3)は固定Qコーダーの状態0-約0.5の確率で符号化されるというものである。

【0320】実施例によっては、エントロピー符号化は各符号化単位の後でリセットされるので、適応可能なコンテキスト(例えば最初のオン・ビットの前の2進値の符号化に用いられるコンテキスト)のための適応化コストが重要である。このコストを最小に保つため、これらコンテキストのための初期状態の集合が、例えば、いくらかのトレーニングデータから計算されるかもしれない

い。

【0321】以下の説明は、係数が18ビットであり、入力データが4レベル分割されていると仮定している。

【0322】符号/絶対値ユニット109の一実施例が図35に示されており、これは入力した係数を符号/絶対値形式に変換する。符号/絶対値ユニット109は18ビットの係数を受け取るように接続され、インバータ2801とマルチプレクサ(MUX)2802を含んでいる。符号/絶対値ユニット109は、有意度表示(例えば5ビット値)、入力係数の仮数(例えば17ビット)、入力係数の符号の1ビット、及びカウンタ2804からの指数(例えば7ビット)を出力する。

【0323】MUX2802は、符号/絶対値ユニット109に直接入力した17ビットの係数と、その17ビットを2の補数器(インバータ)2801により反転したものを受け取るように接続されている。MUX2802の選択入力に与えられる符号ビット(係数のビット17)に基づいて、その2入力中の正量が仮数として出力される。

【0324】<符号化の代案>2元エントロピー符号化器は、コンテキストと符号化すべきビットを与えられる。

【0325】1ビットプレーンずつ符号化する場合、本発明は、ビットプレーン毎の符号化を高速に行うのに適したデータ形式で演算が行われるように、(汎用コンピュータ上で)桁上がり保存型の演算を使う。例えば、そのような構成において、32ビットのプロセッサは、1つの係数全部を同時に計算するのではなく、同じビットプレーンにある32個の係数それぞれの1ビットを同時に計算する。このような実施例によれば、1ビットプレーンずつ符号化する時の速度が上昇する。

【0326】1つの符号化単位が一度に符号化され、符号化単位内の全ての係数がメモリ内にあるので、適応2元エントロピー符号化器が必要とするものを別にすれば、コンテキスト情報の記憶のためのメモリコストは全くない。例えば、Qコーダーは、全てのコンテキストのLPS(最下位シンボル)の2進値と、適応可能な各コンテキストのための現在状態を保存しておく必要がある。Qコーダーは30個の状態を持つので、各コンテキストのために6ビット数(LPS用に1ビット、状態用に5ビット)が必要である。したがって、メモリコストは $1024 \times 5 + 1030 = 6150$ ビットである。

【0327】なお、上に述べた1-パスの実施例の場合には格別の合図情報は必要でない。もし分割レベル数が変数ならば、そのために少なくとも3ビットのヘッダー情報が必要になろう。この実施例に用いられるヘッダー(圧縮後のビットには参入されない)は次のとおりである。

- ・ 幅、2バイト
- ・ 高さ、2バイト

・ 入力画像の1画素あたりのビット、 1バイト

【0328】<メモリ管理>データ全部をメモリに格納するシステム及びデータを通信路で伝送するシステムに関して、1パスシステムにおける符号化データのメモリ管理を提案する。1パスシステムでは、符号化データは”埋め込み因果関係法 (embedded causalfashion) ”でアクセスでき、かつ、有意度の高いデータを損なわずに有意度の低いデータを廃棄できるように格納されなければならない。符号化データは可変長であるので、動的メモリ割り当てを利用できる。

【0329】本発明の一実施例では、埋め込み符号化スキームは18個のビットプレーンを用いるので、18レベルの優先度をデータに割り当てる。1パスシステムの符号化器は、”埋め込み因果関係 (embedded causal 1) ”のものでなければならない。すなわち、あるビットプレーンに対応するイベントの復号に、それにより下位のビットプレーンからの情報を必要としない。一実施例では、あるツリーのビット全部が符号化された後に次のツリーのビットが符号化されるから、有意度の異なるビットは分離されない。内部状態 (internal state) を用いないハフマン符号化器のような符号化器の場合には、このことは問題ではない。しかし、多くの圧縮率の優れた高度な圧縮器は内部状態を利用する。

【0330】これら符号化器に関する当該問題を解決する一方法は、18個の別々の符号化器、多分18個の別々のQ-コーダーチップを用いることである。9個のQ-コーダーチップを使用可能であろう手法が、”Data Compression for Recording on a Record Mediumなる発明の名称で1992年3月17日発行された米国特許第5,097,261号 (Langdon, Jr.) に述べられている。されど優れた方法は、単一の物理符号化器で様々な仮想符号化器を実現するためにパイプライン符号器、例えば、”Method and Apparatus for Parallel Decoding and Encoding of Data”なる発明の名称で1995年1月10日発行の米国特許第5,381,145号に述べられている符号化器を用いる。このような符号化器では、各確率毎に複数のビットジェネレータ状態がそれぞれデータの一部に割り当たられる。例えば、18ビットデータの場合、18個の状態のそれぞれをある1つのビットプレーンに割り当たられる。符号化器内部のシフタのレジスタもデータの各部分に割り当たられる。この符号化器においては、インターリープは行なわれない、すなわち、データの各部分は単にビット詰めされるだけである。

【0331】複数の物理符号化器または仮想符号化器を用いる実施例では、データの各部分にメモリが割り当たられる。圧縮が完了した時に、割り当たられたメモリとその内容を記述する連結リストが結果として得られる。

【0332】メモリがオーバーフローすると、メモリ割

り当てルーチンは重要度の高い情報を重要度の低いデータに上書きさせる。例えば、数値データの最下位ビットが最初に上書きされることになろう。符号化データに加え、メモリの割り当てられた方を記述する情報が格納されなければならない。

【0333】図36は、3つの有意度カテゴリーのための動的メモリ割り当てユニットの例を示す。本発明をいたずらに難解にしないため3カテゴリーしか述べないが、一般的には、8カテゴリー、16カテゴリー、18カテゴリーというように、もっと多数のカテゴリーが用いられるであろう。レジスタファイル (または他の記憶手段) 2901が、各有意度カテゴリー毎のポインタ (現在ポインタ : current) に加え、次の空きメモリロケーションを示すもう一つのポインタ (フリー pointers : free) を保持する。メモリ2902は固定サイズのページに分割される。

【0334】最初、1つの有意度カテゴリーに割り当たられた各ポインタは、あるメモリページの先頭を指し、フリー pointersは次の利用できるメモリページを指す。ある有意度カテゴリーに関する符号化データは、対応したポインタによりアドレスされたメモリロケーションに格納される。そして、このポインタは、次のメモリロケーションを指すようにインクリメントされる。

【0335】ポインタが現在ページの最大値に達した時に、フリー pointersに格納されている次の空きページの先頭のアドレスが、リンクとして現在ページと一緒に格納される。一実施例では、この目的のために、符号化データメモリのその部分または独立したメモリもしくはレジスタファイルが用いられるかもしれない。次に、現在ポインタは次のフリーページを指すように設定される。フリー pointersはインクリメントされる。これらのステップにより、新たなメモリページがある有意度カテゴリーに割り当たられ、かつ、ある共通の有意度のデータを含むメモリページのリンクを得られる結果、復号期間に割り当てる順序を確認することができる。

【0336】メモリの全ページが使用中であり、しかも、メモリ内の最低有意度データより有意度が大きいデータがさらに存在する時には、メモリの再割り当たが行なわれるであろう。そのような再割り当ての3つの手法について述べる。いずれの手法でも、最低有意度データに割り当たれているメモリが、それより有意度の高いデータへ再割り当たされ、もはや最低有意度のデータは保存されなくなる。

【0337】第1の手法は、最低有意度のデータに現在使われているページが単純に、それより有意度の高いデータに割り当たられる。最も一般的なエントロピー符号化器は内部状態情報を利用するので、そのページに前に格納されていた最低有意度のデータは全部失われる。

【0338】第2の手法は、最低有意度のデータに現在使われているページは、それより有意度の高いデータに

割り当てられる。第1の手法と違い、ポインタは当該ページの末尾を指すように設定され、それより有意度の高いデータが当該ページに書き込まれるにしたがい対応ポインタはデクリメントされていく。この手法は、有意度の高いほうのデータがページ全体を必要としないときには、ページの先頭にある最低有意度のデータは保存されるという利点を有する。

【0339】第3の手法は、最低有意度のデータの現在ページを再割り当てるのではなく、最低有意度データの任意のページを再割り当てる。そのためには、符号化データは全てのページが独立に符号化される必要があり、このことは圧縮率の低下を招くかもしれない。また、全てのページの先頭に対応する非符号化データを識別する必要もある。最低有意度データの任意のページを捨てる所以ができるので、より大きな量子化のフレキシビリティを得られる。

【0340】上記第3の手法は、画像の複数領域にわたって一定の圧縮率を達成するシステムで特に魅力的であろう。ある指定した数のメモリページを画像の1つの領域に割り当てる所以ができる。有意度の小さなデータが保存されるか否かを、ある領域において達成される圧縮率に応じて決める所以ができる。なお、ある領域に割り当られたメモリは、非損失性圧縮が必要とするメモリがそのメモリ量より少ないならば、全部は使われない。画像のある領域に対し、ある一定の圧縮率を達成することにより、その画像領域へのランダムアクセスをサポート所以ができる。

【0341】圧縮が完了した時に、そのデータを必要なならば有意度の順に通信路または記憶装置へ送ってよい。そうすれば、様々なリンク(link)とポインタはもう必要ではなく、マルチパスの復号を行ない得る。あるいは、1パス復号の場合には、各有意度毎にデータのポインタを保存できる。

【0342】用途によっては、有意度カテゴリーの中には使われないカテゴリーがあるかもしれない。例えば、16ビットの圧縮器が12ビットの医用画像に対し使われると、ビットプレーン15...12に対応した有意度カテゴリーは使用されないであろう。ページが大きく、かつ、多くの利用されない有意度カテゴリーがある例では、(いくつかのカテゴリーが使われないことをシステムが予め知らない時には)それらの使わないカテゴリーにメモリを割り当てる必要はないのであるから、メモリが浪費されることになろう。このメモリ浪費に対するもう一つの解決策は、各有意度カテゴリー毎のカウントを保持するための小さなメモリ(もしくはレジスタ)を用いることであろう。このカウントは、他のデシジョンが発生する前に発生した"非有意な"デシジョンの数を記録することになろう。不使用の有意度カテゴリーに使われるメモリの代わりに、これらのカウンタを格納するためのメモリが必要になるという代償を払わなければなら

ない。

【0343】システムにおいて利用可能なメモリの全量をより上手に利用するため、各ページにその両端からデータを書き込む機能を利用する所以ができる。全ページが割り当られている時に、一端側に十分な空きスペースのあるページを、その端から利用するように割り当てる所以ができる。ページの両端を利用する機能は、2種類のデータがぶつかるロケーションの管理コストと比較衡量されなければならない。もっとも、一方のデータ種類が重要でなく単純に上書きされてもよい場合は別である。

【0344】<改良>本発明は、一実施例において、小さなメモリバッファで非損失性圧縮をする。本発明は、様々な用途と装置環境に適用可能である。以下に、本発明のシステムを様々な用途及びターゲット装置に対しより柔軟に適応できるようにするためのいくつかの機能を実現する手法を述べる。なお、本発明は、解像度、画素深度、ランダムアクセス、量子化等々の選択を符号化時点でするべきではない。

【0345】<データの取り決め>データの取り決めに関して、画像及び係数データを本発明のシステムと取り決めるためのいくつかのオプションがある。以下に詳しく述べるように、これらのオプションには、符号化単位のタイリング、分割レベル数、ウエーブレット変換フィルタの選択、係数のアラインメントが含まれるが、それらに限定されるものではない。これらオプションのそれぞれを、ユーザー又はシステム設計者の制御するパラメータとしてもよい。

【0346】上に述べたように、一つのパラメータは符号化単位のタイリングであろう。符号化単位の高さ及び幅は、本発明のツリーを考慮して定義される。ランダムアクセスのために、符号化単位毎に符号化データの先頭を、符号ストリーム中のポインタもしくはマーカー又はヘッダ中のポインタにより指定する所以ができる。こうすると、画像のブロックをアクセス可能になろう。

【0347】制御されるであろうもう一つのパラメータは分割レベル数である。分割レベル数が多いほど良い圧縮が得られるので、分割レベル数を変えると圧縮性能が変わる。なお、分割レベル数が増えると必要なラインバッファが増えるので、分割レベル数を変えるとメモリ条件に影響を与える。フル解像度より低い解像度をターゲットにして、それより多くのレベル数が必要なこともある。例えば、オリジナル画像が2000dpiの場合、約36dpiを得るには5レベル分割が必要となる。これにより、伸長してサブサンプリングすることなく、高解像度スキャンをほぼ実サイズでモニタに表示させることができる。

【0348】水平及び垂直パス用のウエーブレットフィルタのタイプがレベル毎に異なってよい。これは、色々なメモリ条件及び圧縮性能を可能にする。なお、レベ

ル数が増えても係数のサイズは増加しない。また、ウエーブレット変換はN次変換であり、また、レベルが増加するほど変換すべきデータが少なくなるので、レベルの増加による演算コストの増加は殆どない。

【0349】<埋め込み符号ストリームのターゲット装置>圧縮符号ストリームには多くの応用目的がある。低解像度であるがフル画素深度のモニターや、フル解像度であるが低画素深度のプリンタ、有限通信路の固定レート・リアルタイム装置、固定サイズの有限メモリ装置へ送ることが可能な符号ストリームを得られると望ましい。同じ符号ストリームでこれらニーズ全てに応えることが要求されるかもしれない。図41は、単一の用途に適用する場合の相対的装置特性の概要を示す。

【0350】<伝送又は復号・符号ストリームパーサ(parser)>符号化器及び復号器に十分な速度を持ちかつ十分なバンド幅を持つ本発明システムは、伸長画像から必要なデータを抽出することができる。さらに、符号化器は、上記装置のどれかのための符号ストリームを生成できる。符号化器において、画像を普通の方法で量子化又はダウンサンプリングすることができる。

【0351】しかし、本発明の1つの長所は、前記装置のどれかのための伸長を行わないで、伝送又は復号の前に構文解析する(parse)ことが可能な符号ストリームを生成できることである。そのようなパーサを図42

(A) 及び (B) で説明する。図42 (A) には、非損失性ビットストリームを受け取って損失性ビットストリームを生成するパーサ3501が表されている。図42 (B) には、損失性ビットストリームを受け取って別の損失性ビットストリームを生成するパーサ3502が表されている。しかし、図42 (B) における出力と入力の間の関係は、本発明がべき等の性質を持つというものであり、これについて以下にさらに詳しく述べる。なお、パーサ3501、3502のいずれの場合も、入力として受け取られるデータのビットレートは出力されるビットレートより大きい。

【0352】<低解像度、高画素深度埋め込みターゲット>ターゲットが低解像度、高画素深度埋め込みターゲットであれば、当該応用を、そのターゲット装置は利用できる空間解像度より低い空間解像度を持つがフル画素深度が要求されると推定する。低解像度、高画素深度埋め込みターゲットの例は各種モニタである。図30に示す符号ストリームを用いる場合、各ビットプレーンは必要なだけの数の高レベル係数について復号される。これには、パーサに各ビットプレーンの切り捨てをさせなければならない。パーサを支援するため、各符号化単位の各ビットプレーンに、切り捨てをしてよい位置を示すマーカ又はポインタを持たせることができる。このような実施例では、2つ以上のターゲット解像度が必要ならば、もっと多くのマーカ又はポインタが必要となる。エントロピー符号化器を次のビットプレーンのためリセッ

トすることができるよう、ビットプレーンは独立に符号化される。

【0353】もう1つの方法は、データを別のやり方で、例えば図31に示すように埋め込むことである。図31において、各符号化単位内のターゲット解像度の係数をまず符号化し、その後に残りの高解像度係数を続ける。この場合、符号化単位あたり必要な切り捨ては1回だけであり、エントロピー符号化器はリセットされる必要がない。マーカ又はポインタにより必要な切り捨て点を指定できる。

【0354】<高解像度、低画素深度埋め込みターゲット>ターゲットが高解像度、低画素深度埋め込みターゲットであれば、当該応用を、そのターゲット装置は利用可能なフル解像度以上の解像度を要求するがフル画素深度を利用できないと推定する。高解像度、低画素深度埋め込みターゲットの例に、ローエンドのプリンタや標準モニタがある（画像が8ビット/プレーン以上のとき）。図30に示す符号ストリームは、このような順序で埋め込まれる。各符号化単位は、適当なビットプレーン数となる点で切り捨てられ、変換は量子化された係数に対しなされる。係数深度と画素深度との間には直接的な関係がある。マーカ又はポインタで所望の切り捨て点を指定できる。

【0355】あるいは、符号ストリームが図31に示すように埋め込まれる場合には、打ち切りを指定するために2つのマーカ又はポインタが、すなわち低解像度ビットプレーン用の1つと、高解像度ビットプレーン用の1つが用いられる。エントロピー符号化器をリセットできるようにするために、2組のビットプレーンが独立に符号化される。

【0356】また、もう1つの方法は、低解像度、高画素深度埋め込みターゲットに関連して述べたように、低解像度の係数の一部又は全部、それに、できれば高解像度係数からのデータを符号化する方法である。そして、後述の補間ウエーブレット変換を行う。

【0357】<固定レート埋め込みターゲット>ターゲットが固定レート埋め込みターゲットであれば、当該応用を、束縛通信路を使い、しかもリアルタイムの一定画素出力を維持しなければならないと推定する。この場合、時間局的に符号ストリームのデータレートが最大（圧縮率は最小）になる。この目標とするところを達成するため、ターゲット装置で利用できるバッファリング量に基づいて符号化単位が選ばれる。これにより、平均的圧縮率を達成できる場所が決まる。そして、各符号化データの許容量を超えるデータは切り捨てられる。

【0358】なお、符号ストリームのデータレートが通信路の最大バンド幅を超えないければ、画像は非損失で復元される。このことは、他の固定レートシステムには当てはまらない。

【0359】<固定サイズ埋め込みターゲット>ターゲ

ットが固定サイズ埋め込みターゲットであれば、当該応用を、固定サイズのフレームバッファが圧縮画像データに利用できると推定する。これは、固定レートの応用とは違い、局所的でなく画像全体にわたり平均した、ある最小の圧縮率を必要とする。ここで固定レートの方法を利用できることは勿論であるが、局所的にではなく画像全体にわたり平均をとるという考え方を用いることにより、ビット割り当てと画像品質を改善することができる。

【0360】符号化単位が画像全体を包含するならば、バッファから溢れるデータを切り捨てるることは当たり前であろう。符号化単位が画像全体より小さく、かつ、全ての符号化単位が同一ビット数に切りつめられるときには、その切り捨てにより一様に最低有意度レベルが除去される保証はない。簡単な解決法は、符号化時に（又はその後の構文解析時に）、各有意度レベルが符号ストリームに与えた符号化ビットの数を、符号化単位毎に、又は包括的に、あるいは、その両方で、記録することである。この記録は簡単なカウンタを使って行うことができる。これらの数は、ヘッダに記録され、伝送時又は格納時に各符号化単位の切り捨て方法を決めるために利用できる。ヘッダは有意度レベルと、それに対応したビット数を保持している。ヘッダは、この情報を、ストリームに含まれている符号化単位それぞれ毎に保持してもよい。切り捨てる場所を決定する際、影響を各符号化単位に対し同一にする。例えば、メモリが制約されたため、1.5有意度レベル分を切り捨てなければならないときには、各符号化単位から1.5有意度レベル分が切り捨てられる。こうすることで、切り捨ての影響を符号化単位に均一に分散させることができる。

【0361】符号化時も同様に、固定サイズ圧縮画像を得ることができる。メモリは各有意度レベル用のセグメントに分割される。メモリが溢れそうになると、優先度レベルの低いセグメントはそれより優先度レベルの高いデータにより上書きされる。

【0362】なお、圧縮データがメモリバッファから溢れなければ、その画像は非損失で復元される。このことは他の固定サイズシステムには当てはまらない。

【0363】<ウエーブレット変換による補間>ウエーブレットを用いて、画像をより高解像度になるよう補間することができる。その結果は視覚的に2重3次スプライン法に極めて近い。圧縮データが既にウエーブレット係数の形であれば、補間のための実質的な計算量の増加は2重3次スプラインより少ない。

【0364】Nレベル分割の係数を全て利用できるとしよう。ゼロの埋め込み(padding)又は他の方法により新たな最低レベルの係数を生成し、そしてN+1レベルのウエーブレット復元を実行することにより、新たに得られた画像はオリジナル画像を2:1に補間した画像である。

【0365】この方法は、埋め込み符号ストリーム用ターゲット装置、特に高解像度、低画素深度用ターゲット装置の存在するシステムにも利用できる。符号化単位は、低解像度係数だけが存在するよう（あるいは、低解像度係数と、高解像度係数の一部又は全部の数ビットだけが存在するよう）切りつめられる。これら係数は、それより高い解像度に埋め込まれ、そして復元が実行される。

【0366】<通信路の利用>データがメモリに格納されるのではなく通信路で伝送され、かつ、固定サイズのメモリページが利用される（1有意度カテゴリーにつき1ページだけ必要とされる）システムにおいて、1つのメモリページが一杯になった時に、そのページは通信路で伝送され、そして、メモリロケーションを伝送されるとすぐに再び使用できる。用途によっては、メモリのページサイズは通信路で使われるデータパケットのサイズまたはパケットサイズの倍数とすることができる。

【0367】通信方式によっては、例えばATM（非同期転送モード）では、パケットにプライオリティを割り当てることができる。ATMは、2つのプライオリティレベル、つまりプライマリとセカンダリを有する。セカンダリパケットは十分なバンド幅を利用できるときのみ伝送される。閾値を使って、どの有意度カテゴリーがプライマリであるか、どの有意度カテゴリーがセカンダリであるかを判断することができる。もう1つの方法は、符号化器で閾値を利用して、閾値より有意度の小さい有意度カテゴリーを伝送しないようにすることであろう。

【0368】このように、本発明のメモリマネージャの一実施例は固定サイズのメモリへの圧縮データの格納を制御する。すなわち、このメモリマネージャは、圧縮データを異なった有意度レベルに分ける。メモリが一杯になった時に、有意度の低いデータは有意度の高いデータによって上書きされる。

【0369】限られた量のバッファメモリを使って通信路を管理するために（例えば固定レート）、本発明の一実施例では、十分なバンド幅を利用できるときにはデータは全て伝送されるが、十分なバンド幅を利用できないときには、有意度の低いデータは捨てられ有意度の高いデータだけが伝送される。

【0370】図37は通信路マネージャを利用するシステムを表している。図37において、ウエーブレット変換3001は係数を生成する。これら係数はコンテキストモデル3002により処理される。コンテキストモデル3002は、バッファメモリを含む通信路マネージャ3003と接続される。通信路マネージャ3003は有限バンド幅の通信路3004に接続される。

【0371】通信路マネージャ3003は通信路3004へのデータの出力レートを制御する。通信路マネージャ3003は、そのバッファメモリにデータが入力され

る時に、そのデータ量が通信路3004のバンド幅を超えるか判定する。そのデータ量がバンド幅を超えないときには、通信路マネージャ3003はそのデータを全て出力する。他方、バッファメモリに受け取ったデータ量が通信路のバンド幅を超えるときには、通信路マネージャ3003はバッファメモリ内のデータを通信路3004のバンド幅に整合するように捨てる。

【0372】通信路3004は、そのバンド幅を通信路マネージャ3003に指示するかもしれない。もう1つの実施例では、通信路マネージャ3003は、所定の1単位（例えば1パケット）のデータを通信路3004を通じ送信するのに要した時間の長さに基づいて、通信路3004のバンド幅を動的に決定するであろう。すなわち、通信路バンド幅を、望むならば動的なものとして扱うことができる。

【0373】一実施例では、通信路マネージャ3003はタイル（又はバンド）に分割された画像を処理する。これは”有意度よりタイルを優先する”スキームであり、これに対し固定サイズのメモリマネージャではタイリングと有意度は多少独立している。各タイルは別々に符号化され、有意度レベル別及び固定サイズページ別に分けられる。したがって、各タイル毎の符号化データは全て一緒にまとめられる。各タイル内の符号化データは有意度レベル別にタグ付けされる。

【0374】一実施例では、通信路マネージャ3003内のバッファメモリは少なくとも、通信路のパケットサイズの2倍（あるいは3倍）、タイルの予想圧縮データサイズの数倍（もしかすると4倍）のサイズである。

【0375】バッファメモリの一定の最大量が1つのタイルに割り当てられる。この最大量は通信路のバンド幅と整合がとられる。バッファメモリは固定サイズのセグメントに分割され、必要な時に割り当てられる。メモリ使用量が許容最大値に達すると、固定サイズメモリシステムの管理の場合と同様、セグメントは再割り当てされる。

【0376】図38はバッファメモリの使用例を示す。図38において、循環バッファメモリは複数の固定サイズのセグメント3101を持ち、それらは通信路入出力のための複数の固定サイズのパケット3102に分割される。図示のように、データの様々なタイルがメモリの同じパケットを使用する。一実施例では、異なったタイルは異なった有意度レベルを意味する。パケットサイズの大きさのバッファ空間が使用されると、通信路マネージャ3003はデータを通信路3004（図37）へ出力するようコンテキストモデルに指示する。図示のように、タイルN-2と、タイルN-1の一部分が現在パケットとして出力されよう。このように、通信路のバンド幅と整合するよう、パケットサイズ量のメモリが割り当てられてデータが入れられる。

【0377】バッファが満杯にならなければ、余ったメ

モリをその後のタイルに利用できる。一実施例では、次のブロックに比べ圧縮しにくい領域の始まりで重要なタイルが区切られないようにするため、余ったメモリの数分の1（1/2, 1/3等々）しか次のタイルに用いられない。

【0378】本発明の通信路マネージャは、データを一定時間内に伝送するしかない場合に利用し得る。そのような通信路マネージャを用いると、データ伝送は、その複雑さに関係なく、その時間内に発生する。というのは、データはその有意度に基づいて埋め込まれているからである。

【0379】＜通信路マネージャの他の実施例＞本発明の通信路マネージャの1つの目標は、メモリ消費を最小にすることである。一実施例において、通信路マネージャがバッファメモリを含まない場合、以下のものを採用してよい：

各符号化単位毎

各符号化単位毎の各ビットプレーン毎

各符号化単位毎の各ビットプレーン毎の各周波数毎

各符号化単位毎の各ビットプレーン毎の各周波数毎の各空間位置毎。

【0380】一実施例では、符号化器（ある既知の状態に設定されている）は各バンドの初めでリセットされる。一実施例では、4レベル分割の場合、バンドメモリを縮小したいならば、1つのバンドは16ラインからなる。

【0381】図39は上記方法によるビットストリームを表している。図39において、ビットストリームは固定サイズのセグメントに分割される。このセグメントは、通信路パケット、ディスクのセクタ、その他、通信路用の合理的なバッファ量なら何でもよい。なお、この分割は、符号化期間中の論理的な分割にすぎない、すなわち、望むならば符号化器はバッファリングせずに出力してもよい。各固定サイズセグメントは、同セグメント中の最高有意度のデータの表示を含んでいる。

【0382】セグメントの構造が図40に示されている。図40において、1セグメントのビットストリームは符号化データ3301、オプションのポインタもしくはID3302、及び当該セグメント中の最高の有意度のデータのレベル3303からなる。一実施例では、ビットフィールド3303は2～5ビットからなる。その最高有意度レベルが0（最高の有意度レベル）ならば、セグメントの末尾から前のMビットは同セグメント内のレベル0のデータが始まる位置を示すポインタである。なお、最初のセグメントのデータが完全に符号化されたデータならば、オーバーヘッドは不要である。

【0383】一実施例では、各バンドの開始点は、JPEG規格に用いられているものと同様の再スタート・マークを利用して識別される。しかし、使用されるマークは、符号化中に発生頻度が最小のシンボルを用いる

べきである。

【0384】再び図39について検討する。一部の有意度レベルだけ（たぶん最高有意度レベルだけ）を伸長したいとする。伸長は、最初のセグメントから始まる。セグメント2について、"セグメント中で最も有意度の高いデータのレベル"がチェックされ、そして恐らく、セグメントに含まれている最も高い有意度が伸長されるレベルより低い場合にはセグメント全体をスキップできる。3番目のセグメントについては、ポインタを利用してバンド2の先頭が見つけられ、バンド2の伸長を開始できる。

【0385】なお、セグメント内の最も有意度の高いデータ全部を必ず得られるようにするために、セグメント全体を伸長する必要がある、特に、そのセグメントに入っているものが1バンドより多い場合にはそうであろう。

【0386】予め決めた数の有意バンドだけを選択的に伸長すれば、プレビュー画像が得られるであろう。これは、データが埋め込み形式であって非損失性データの損失性データが必要とされるときに好都合であろう。

【0387】望ましいアクセス及び量子化の可能性、並びに、バンド全体を伸長するための時間が重要なか否かによって、セグメントの最後のオプションのポインタ又はIDに次のものを入れることができる：

- ・固定サイズメモリ管理のための次セグメントポインタ。
- ・セグメントのID又は含まれているバンドのID。
(通信路量子化をサポートする、例えばセグメント2が除かれたか否かを表示する)。
- ・セグメントに含まれている異なったバンドの数（又は、少なくとも、1つのバンドが2セグメント以上を含んでいることを表示するビット）。(所要データを伸長した後はセグメント全部は伸長しないことをサポートする)。

【0388】 512×16 画素の1バンドあたりのオーバーヘッドの例であるが、8ビット画像が2:1に非損失性圧縮され、セグメントのサイズが512バイトであるとする。なお、1バンドは典型的には8セグメントまで圧縮する。32有意度レベルの場合、5ビットのタグが使われる。ポインタがバイト境界にあるとすれば、9ビットのポインタが用いられる。したがって、(32圧縮ビット+49)あたり49ビットのオーバーヘッドがあり、これは合計0.15%に相当する。

【0389】<略非損失性>略非損失性の圧縮という概念は、復元画素値の絶対誤差をベースにしている。よって、絶対誤差が1の略非損失性圧縮画像の場合、その伸長画像中のどの画素値も元の画素値との違いが画素値の1単位を超えない。これは、画素深度や画像のダイナミックレンジに依存しない絶対的な定義である。このようなシステムにとって、明白かつ、いくつかの合理的な仮

定の下で最適なことは、圧縮／伸長部を非損失性のままにしておき、略非損失性を達成するための前処理スキーム及び後処理スキームを用いることである。この方法は当該実施に採用された。

【0390】絶対誤差がeの略非損失性圧縮画像は、2e+1個の連続整数をすべて、それらの中央の整数に写像する量子化法によって得られる。例えば、誤差=3の場合、画素値は、0~6は3に写像される、7~13は10に写像される、というように量子化される。この量子化画像それ自体は、変換ベースの圧縮システムには適さない。そこで、量子化画像は、それよりもダイナミックレンジ又は深度が低い画像（浅い画像と呼ぶ）へ1対1（非損失）写像される。この写像は、中央値（代表値）を、その順序を変えずに連続整数へ写像することによってなされる。数学的には、ある画素値xが与えられると、この画素値は次のように量子化される。

【0391】

【数34】

$$q(x) = \left(\left\lfloor \frac{x}{2e+1} \right\rfloor + 1 \right) e$$

【0392】代表値の浅い画像値への1対1写像は次のとおりである。

【0393】

【数35】

$$p(x) = \left\lfloor \frac{x}{e} \right\rfloor - 1$$

【0394】浅い画像値を再び代表値へ写像する、1対1写像pの逆写像は次のとおりである。

【0395】

【数36】

$$p^{-1}(x) = e \cdot (x + 1)$$

【0396】量子化($q(x)$)とその後の浅い画像値への写像($p(x)$)が、非損失性圧縮に先立つ前処理演算である。浅い画像値から代表値への写像が、非損失性伸長の後に続く後処理演算である。

【0397】変換領域量子化も利用できる。多くの係数が、変換の多くのレベルへのピーク誤差伝搬に影響を及ぼす。子のない高域通過係数については、ピーク誤差への影響を調べることは容易である。

【0398】1次元信号を最大ピーク誤差±Eで符号化することを考える。これは、最精細ディテールの高域通過係数を±2Eに量子化することにより達成される。2次元信号の場合、高域通過フィルタが2回かけられるので、最精細ディテールのHH係数を±4Eに量子化する。

【0399】入力画像の量子化を利用する方法に代わる方法は、エントロピー符号化器に対するデシジョンを制

御することである。一例は以下のとおりである。各係数について、係数をゼロに設定しても、その係数に影響されるどの画素の誤差も最大誤差を超えないければ、その誤差はゼロに設定される。実施例によっては、特定の係数だけ、例えば子を持たない A C 係数だけが調べられる。係数は、1 度に 1 つずつ調べる念入りな方法で調べてもよい。別の方では、係数の小グループを考えて、グループの可能な限り大きな部分集合をゼロにすることに決めてよい。

【0400】上述のように、量子化は埋め込み作用によりなされ、そして RMS E のような定量的基準に関し性能を最大にするように最適化される。一実施例では、ハフマン視覚系に関し改善された結果を得るよう、いくつかの係数の量子化が行われる。このような場合、埋め込みスキームは殆ど変更する必要がない。例えば、係数は、それら係数間の関係を 2 分の 1 だけ変更し、及び／又は、グレイコードのような別種の記数法により数を表現するためシフトされる。

【0401】本発明の圧縮ウェーブレットシステムは、画像編集にも有効であろう。従来技術では、画像処理機能をフル解像度の印刷画像に適用すると、時間がかかり対話的処理が困難になる。

【0402】一実施例では、画像編集システムは圧縮されたタイルを保存するならば、ユーザが評価するためのスケール画像（その低域通過）に対し極めて素早く操作を施すことができるであろう。素早くできるのは、表示されている画素だけが操作されるからである。それは最終結果の近似にすぎない、というのは、実際のフル解像度画素が outcome に影響を及ぼすためである。したがって、ユーザーは画像の様々な部分をクローズアップするであろう。ユーザがそうした時に、画像編集システムは、画像の該当部分に操作を施す。これを容易にするため、圧縮された係数、並びに、どの処理操作が施されたか及び施される必要があるかについての情報を含むツリーが保存される。

【0403】一実施例では、定義されたウインドウ内では非損失性圧縮を、画像のその他の部分に対しては損失性圧縮を可能にするように、有意度レベルが再定義される。そのウインドウは固定されても、ユーザにより選択できるようにしてもよい。有意度の違う複数のウインドウがあつてもよい。一実施例では、ウインドウは 48 × 48 ブロック程度の大きさである、ただし、ウインドウをずっと小さく、2 画素程度まで小さくすることも可能である。

【0404】その可能な応用は衛星イメージであり、衛星が非損失性ウインドウをデータに対し使うことにより、統計的研究が J P E G の人為構造(artifacts)によって搅乱されず、また損失性圧縮が非損失性圧縮より遙かに広い視野を可能にする。

【0405】一実施例では、ユーザは画像上に任意の枠

(box) を描き、その枠内のデータの相対的有意度を指定する。枠が描かれると、ソフトウェアがその枠のサイズを、所要の制約条件に合致しかつ当該ユーザ枠を包含する最小サイズまで増大させる。ファイル・ヘッダは使用された枠と有意度レベルに関する情報を保持することになる。そして符号化器及び復号器は、符号化／復号の実行時に、その重要な枠内の係数に高い解像度を与えるであろう。衛星イメージの場合、重要なウインドウは恐らく前もって定義される。

【0406】<ベキ等動作> 損失性圧縮装置の場合、一般に、D C D C I = D C I のようにベキ等動作である。ただし I は画像、C は圧縮操作、D は伸長操作である。本発明において、データが X ビットに圧縮されてから伸長された時に、それを X ビットに再圧縮して元の X ビットを得ることができるはずである。埋め込みシステムについてはさらに強力なベキ等元のバーションがある。一実施例では、画像を X ビットに圧縮し、伸長し、そして Y (ただし Y < X) ビットに再圧縮したものは、元の画像を Y ビットに圧縮したものと同じである。

【0407】このことが重要なのは、圧縮及び処理が画像を元画像から大きく変化させるからである。圧縮装置がベキ等であれば、圧縮伸長サイクルを繰り返してもデータに悪影響を及ぼさない。本発明においては、データが何回、同じ圧縮率で圧縮伸長されるてもかまわない。また、バーサに対する損失性入力がさらに量子化されると、非損失性入力を用いる場合と同じ結果を得られる。このように、本発明は、ウェーブレット変換、コンテキストモデル及びエントロピー符号化器を含み、情報を取り除いても前の係数の記述が変わらないような順序で係数が記述され格納される、変換ベースのベキ等システムを構成する。

【0408】<用途> 本発明は多くの用途に利用できる。そのような用途のいくつかを例として以下に述べる。つまり、解像度が高く画素深度が大きいハンドエンドの用途及び人為構造(artifact)を許容しない用途に、本発明を利用できる。本発明によれば、ハンドエンドの用途は高品質環境で最高品質を維持でき、同時に、バンド幅、データ記憶又は表示機能がさらに制限された用途でも同じ圧縮データを利用可能である。これはまさに、ウェブ・ブラウザ (web browser) のような現代の画像応用分野に一般に要求される装置独立な表現である。

【0409】高画素深度 (10 ビット～16 ビット／画素) の画像に対する本発明の優れた非損失性圧縮性能は、医用画像にとって理想的である。非損失性圧縮のみならず、本発明は、ブロックベース圧縮装置に知られている人為構造(artifacts) の多くがない真の損失性圧縮装置である。本発明を利用することにより引き出される損失性人為構造(artifacts) は、急峻なエッジに沿う傾向があり、ハフマン視覚系の視覚マスキング現象によって隠されることが多い。

【0410】本発明は、画像が非常に高解像度で高画素深度であることの多いプリプレス(pre-press)業に関連した用途に利用できる。本発明のピラミッド分割によれば、プリプレス・オペレータが(モニタ上の)画像の低解像度損失性バージョンに対し画像処理操作を行うのが容易である。操作が終わったならば、同じ操作を非損失性バージョンに対して実行できる。本発明は、圧縮しないと伝送に必要な時間が非常に長いファクシミリ文書の用途にも適用可能である。本発明は、様々な空間解像度及び画素解像度のファクス装置より、非常に高い画像を出力できる。この応用では伝送時間が割高なものがあるので、本発明の補間機能は有益である。本発明は、圧縮を必要とする、特に記憶容量が増すほど圧縮を必要とする画像アーカイブシステムに利用することもできる。本発明の装置独立な出力は、バンド幅、メモリ及びディスプレイが異なる資源を持つシステムにより画像アーカイブシステムをアクセスでき有益である。本発明のプログレッシブ伝送機能は、ブラウジング(browsing)のためにも有益である。最後に、画像アーカイブシステムの出力装置用に望ましい非損失性圧縮が本発明により提供される。

【0411】本発明の非損失性又は高品質損失性データストリームの階層プログレッシブ性により、本発明はワールド・ワイド・ウェブ(world wide web)用に、特に装置独立性、プログレッシブ伝送、高品質、及びオープン規格が必須な場合に理想的である。

【0412】本発明は、衛星画像、特に高画素深度及び高解像度になる傾向のある衛星画像にも適用できる。さらに、衛星画像用途は通信路のバンド幅が制限される。本発明はフレキシビリティがあり、また、プログレッシブ伝送特性があるので、本発明を利用して人間による画像のブラウジング又はプレビューを可能にすることができる。

【0413】データに対し非損失性ウインドウを使うことにより、統計的研究がJPEG人為的構造(artifacts)により搅乱されない衛星もあるが、損失性圧縮によれば非損失性圧縮より遙かに広い視野が可能である。

【0414】一実施例では、定義されたウインドウ内では非損失性圧縮を、画像のその他の部分に対しては損失性圧縮を可能にするように、有意度レベルを再定義できる。そのウインドウは固定されても、ユーザにより選択できるようにしてもよい。有意度の違う複数のウインドウがあつてもよい。一実施例では、ウインドウの最小単位は恐らく 48×48 ブロックとされる、ただし、2画素程度まで精密にすることも可能である。こうすれば、非損失性領域に近い精密な画素を得られるであろうが、非損失性領域が目立たなくなるかもしれない。

【0415】その一方法は、ユーザに画像上に任意の枠(box)を描かせて、その枠内のデータの相対的有意度を指定させることにより実現されるかもしれない。ソフト

ウェアは、その枠を、所要の制約条件に合致しかつ当該ユーザ枠を包含する最小サイズまで拡大させることになる。ファイル・ヘッダは使用された枠と有意度レベルに関する情報を保持することになる。そして符号化器及び復号器は、符号化/復号の実行時に、その重要な枠内の係数に高い解像度を与えるであろう。衛星イメージの場合、重要なウインドウは恐らく前もって定義される。なお、これは衛星画像以外の用途にも適用可能である。

【0416】ATMネットワークのような"固定レート"でバンド幅が制限される用途は、データが利用可能なバンド幅をオーバーフローしたときにデータを減少させる手段を必要とする。しかしながら、十分なバンド幅があるときには(あるいは圧縮率が十分高いときには)、品質上の不利益があつてはならない。同様に、コンピュータや他の画像装置におけるメモリが制限されたフレーム記憶装置のような"固定サイズ"の用途も、メモリが満杯になったときにデータを減少させる手段を必要とする。繰り返すが、適当なメモリ量に非損失圧縮することが可能な画像に対して不利益があつてはならない。

【0417】本発明の埋め込み符号ストリームは、これら両方の用途にかなう。埋め込みは、損失性画像の伝送又は記憶のため符号ストリームが切り捨てられることを暗黙的に許す。切り捨てが必要でなければ、画像は損失なしに届く。

【0418】要するに、本発明は、单一連続階調画像圧縮システムを提供する。本発明のシステムは、同じ符号ストリームに対して非損失性かつ損失性であり、埋め込みの量子化(符号ストリームに必然的に含まれる)を利用する。本発明のシステムはまた、ピラミッド形であり、プログレッシブであり、補間手段を提供し、かつ、実施が容易である。したがって、本発明はフレキシブルな"装置独立な"圧縮システムを提供する。

【0419】統合された損失性かつ非損失性圧縮システムは非常に有用である。同じシステムで最新の損失性及び非損失性圧縮を実行でき、その上、同じ符号ストリームである。この応用は、画像の非損失性符号を保存するか切りつめて損失性バージョンにするかを、符号化中、符号ストリームの格納又は伝送中あるいは復号中に、決定することができる。

【0420】本発明により提供される損失性圧縮は、埋め込み量子化によって達成される。すなわち、符号ストリームは量子化を含んである。実際の量子化(又は視覚的有意度)レベルは、復号器又は通信路によって決まり、必ずしも符号化器によって決まるものではない。バンド幅、記憶及びディスプレイ資源が許すなら、画像は非損失で復元される。そうでなければ、画像は、最も制限された資源によって必要とされる量だけ量子化される。

【0421】本発明に用いられるウェーブレットはピラミッド型であり、差分画像のない、画像の2分の1ずつの分割が実行される。これは階層分割より明確である。画像のブラウジング(browsing)のため又は低解像度装置による表示のために縮小画像(thumbnails)を必要とする用途に、本発明のピラミッド性は理想的である。

【0422】本発明における埋め込みの使い方はプログレッシブである、つまりビットプレーン別であり、MSBの後に下位ビットが続く。具体的には本発明はウェーブレット領域においてプログレッシブであるが、空間領域及びウェーブレット領域の両方ともプログレッシブに分割可能である。プリンタのような、空間解像度はあるが画素解像度は低い用途にとって、本発明におけるビットのプログレッシブな順序づけ理想的である。これらの特徴を同一符号ストリームに対し利用できる。

【0423】本発明の構造の一つの特長は、計算効率のよい補間の方法を提供することである。高い解像度が必要なときには、利用できるウェーブレット係数に基づき高域通過係数を補間することができ、そして本発明の逆ウェーブレットが実行される。この方法は、見かけは2重3次スプラインと競合するが、本発明の変換については計算の集中が遙かに少ない。

【0424】本発明はベキ等である、つまり、画像を損失性の形に伸長し、そして同じ符号ストリームへ再圧縮することができる。この特長は、ブラウジング、フィルタ処理又は編集がある用途において、圧縮・伸長サイクルの繰り返しを可能にする。

【0425】本発明は、ソフトウェアでもハードウェアでも比較的容易に実施できる。ウェーブレット変換は、高域通過係数と低域通過係数の各ペア毎の4つの加算／減算操作と、いくつかのシフトだけで計算することができる。埋め込みと符号化は、単純な”コンテキストモデル”と2元”エントロピー符号化器”によって遂行される。このエントロピー符号化器は、有限状態マシン又は並列符号化器によって実現できる。

【0426】以上、本発明の実施例に関して説明したが、本発明はそのような実施例に限定されるものではなく、様々な変形及び修正が許されることは、以上の説明から当業者には明白であろう。

【0427】

【発明の効果】以上詳細な説明から理解されるように、本発明によれば、従来技術の問題点を改善した効率的な符号化及び復号の方法及び装置を提供することができ、効率的な非損失性又は損失性のデータ圧縮／伸長システムを実現可能である。

【図面の簡単な説明】

【図1】本発明の符号化システムの符号化部の一実施例のブロック図である。

【図2】本発明の係数データの順序付け／モデリングの一実施例のブロック図である。

【図3】ウェーブレット分析／合成システムのブロック図である。

【図4】非オーバーラップの最小長可逆フィルタを用いてフィルタリングするための変換システムのフォワード変換とリバース変換を説明する。

【図5】2-D可逆フィルタの2実施例を示すブロック図である。

【図6】4レベル分割の第1レベル分割結果を示す図である。

【図7】4レベル分割の第2レベル分割結果を示す図である。

【図8】4レベル分割の第3レベル分割結果を示す図である。

【図9】4レベル分割の第4レベル分割結果を示す図である。

【図10】前後2レベル間の親子関係を示す図である。

【図11】3レベル・ピラミッド変換のブロック図である。

【図12】2次元2レベル変換のブロック図である。

【図13】マルチ解像度分割を行なう1次元フィルタを示すブロック図である。

【図14】本発明の可逆ウェーブレットを利用するシステムのブロック図である。

【図15】本発明の可逆ウェーブレットを利用する強調及び分析システムのブロック図である。

【図16】本発明における係数サイズを示す図である。

【図17】本発明における係数アラインメント用の周波数帯域に対する乗数の例を示す図である。

【図18】本発明の符号化プロセスの一実施例を示すフローチャートである。

【図19】本発明の復号プロセスの一実施例を示すフローチャートである。

【図20】本発明のモデリングプロセスのフローチャートである。

【図21】本発明のフォワード・ウェーブレットフィルタの一実施例を示すブロック図である。

【図22】本発明のリバース・ウェーブレットフィルタの一実施例を示すブロック図である。

【図23】いくつかの変換の係数レンジを示す図である。

【図24】ルックアップテーブルを用いるコンテキストモデルの一実施例を示すブロック図である。

【図25】ルックアップテーブルを用いるコンテキストモデルの一実施例を示すブロック図である。

【図26】ウェーブレット分割段階の例を示す図である。

【図27】符号化単位を示す図である。

【図28】TS-変換の垂直パスを示す図である。

【図29】バッファリング及び係数計算を示す図である。

【図30】符号ストリーム構造の一例を示す図である。

【図31】低解像度ターゲット用符号ストリーム構造の一例を示す図である。

【図32】係数(画素)間の隣接関係を示す図である。

【図33】コンテキストモデルの例を示す図である。

【図34】本発明のコンテキストモデルの一実施例のブロック図である。

【図35】本発明の符号／絶対値ユニットの一実施例のブロック図である。

【図36】1パス動作のための符号化データメモリの動的割り当てを示す図である。

【図37】通信路マネージャの一実施例を説明するためのブロック図である。

【図38】本発明におけるメモリ利用法を説明するための図である。

【図39】本発明におけるビットストリームを説明するための図である。

【図40】セグメントの構造を示す図である。

【図41】パラメータ空間とターゲット装置を対比して示す図である。

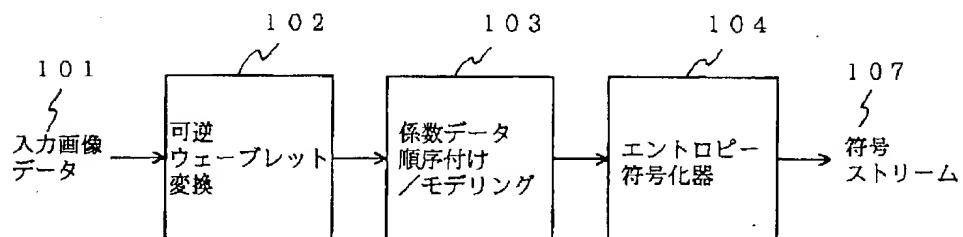
【図42】本発明のパーサー(parser)の実施例を説明するための図である。

【符号の説明】

101 入力画像データ
 102 可逆ウェーブレット変換ブロック
 103 係数データ順序付け／モーデリングブロック
 104 エントロピー符号化器
 106 符号ストリーム
 109 符号／絶対値ユニット
 201 符号／絶対値フォーマッティングユニット
 202 統合空間／周波数(JSF)コンテキストモデルブロック
 211, 212 フィルタ
 213, 214 サブサンプリング・ブロック
 217, 218 アップサンプリング・ブロック
 219, 220 フィルタ
 401, 402 1-Dフィルタ
 403 2-D丸め操作
 701, 702 フィルタユニット
 703, 704 サブサンプリングユニット
 705, 706 フィルタユニット
 707, 708 サブサンプリングユニット
 709, 710 フィルタユニット
 711, 712 サブサンプリングユニット
 721, 722 フィルタユニット
 723, 724 サブサンプリングユニット
 725, 726, 743 フィルタユニット
 728, 729, 730, 744 サブサンプリングユニット

731, 732 フィルタユニット
 733, 734 サブサンプリングユニット
 735～738 フィルタユニット
 739～742 サブサンプリングユニット
 761～766 1-Dフィルタ
 1001 分割ユニット
 1002 圧縮器
 1003 伸長器
 1004 逆分割ユニット
 1101 分割ユニット
 1102 分析ユニット
 1105 分割ユニット
 1106 強調処理及びフィルタ処理ユニット
 1107 逆分割ユニット
 1601 加算器
 1602 減算器
 1603 1ビットシフトブロック
 1604 2ビットシフトブロック
 1605 減算器
 1606 加算器
 1607 2ビットシフトブロック
 1608 加算器
 1701 減算器
 1702 2ビットシフトブロック
 1703 1ビットシフトブロック
 1704 減算器
 1705 加算器
 1706 減算器
 1707, 1708 クリップユニット
 1709 加算器
 1901 ルックアップテーブル(LUT)
 1902 符号化／復号ブロック
 1903 ルックアップテーブル(LUT)
 1904 符号化／復号ブロック
 2700 コンテキストモデル
 2701 ヘッドビットブロック
 2702 符号ビットブロック
 2703 テールビットブロック
 2801 インバータ
 2802 マルチプレクサ(MUX)
 2804 カウンタ
 2901 レジスタファイル
 2902 メモリ
 3001 ウエーブレット変換
 3002 コンテキストモデル
 3003 通信路マネージャ
 3004 通信路
 3501, 3502 パーサ

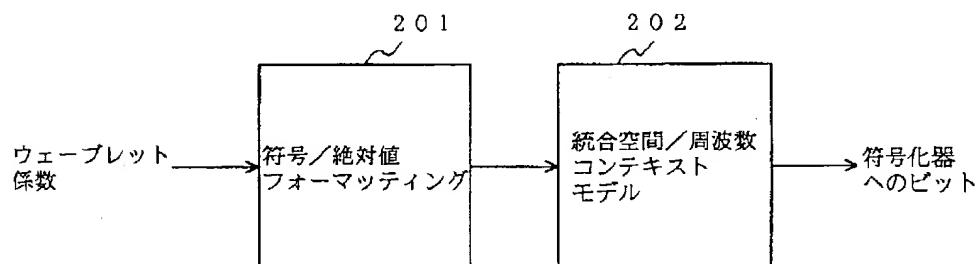
【図1】



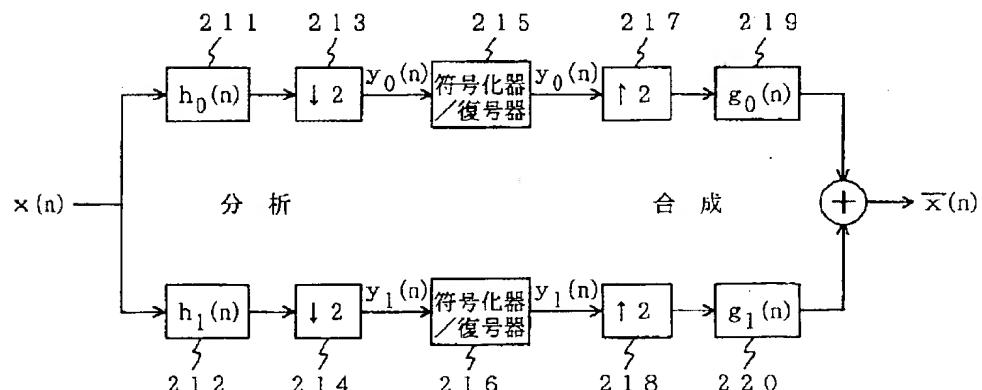
【図3-2】

NW	N	NE
W	P	E
SW	S	SE

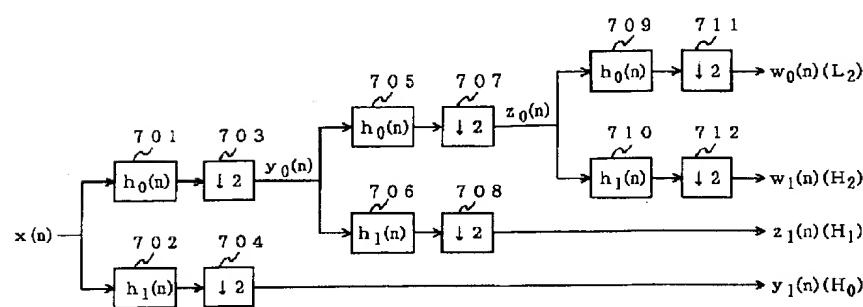
【図2】



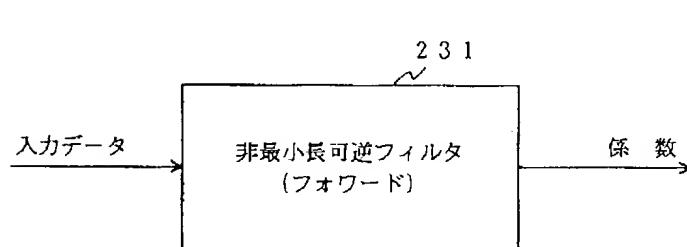
【図3】



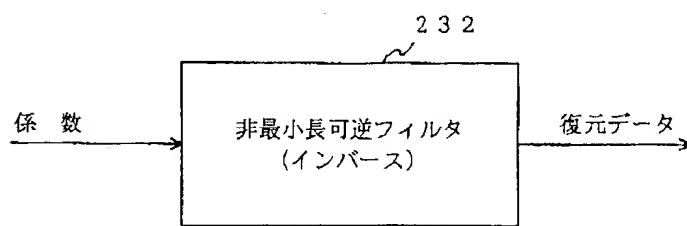
【図1-1】



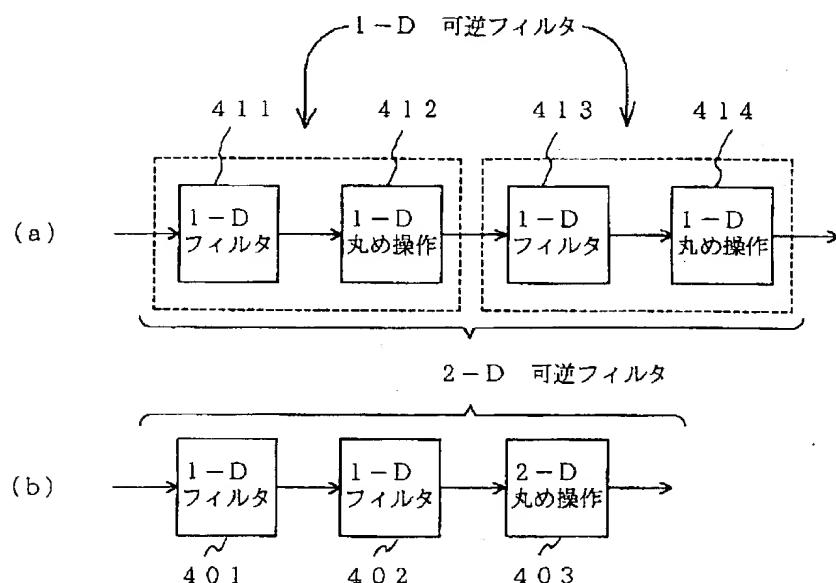
【図4】



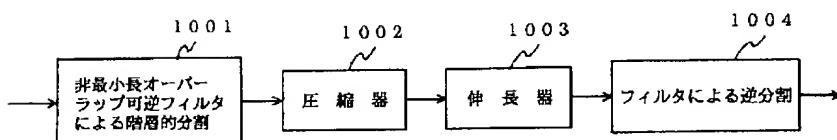
b	$b+2$	
$b+2$	$b+4$	
		$b+2$
		$b+4$



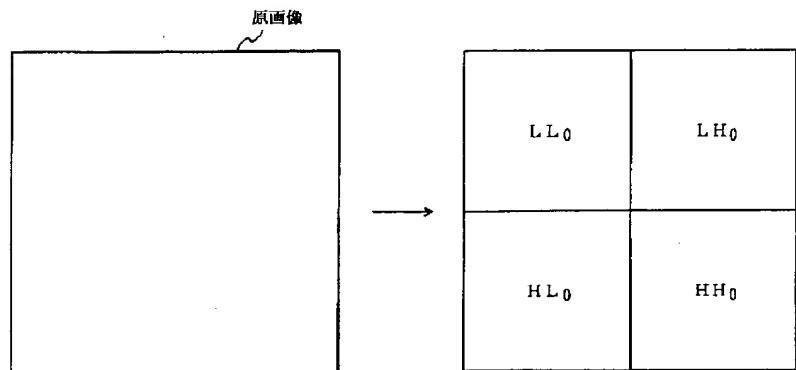
【図5】



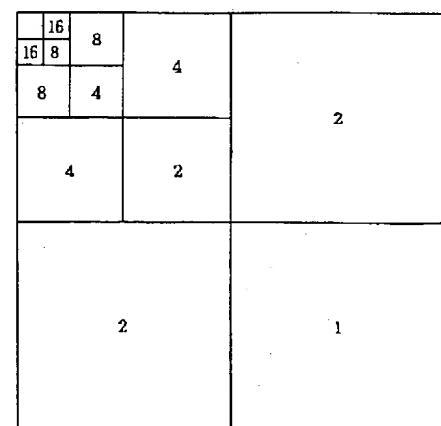
【図14】



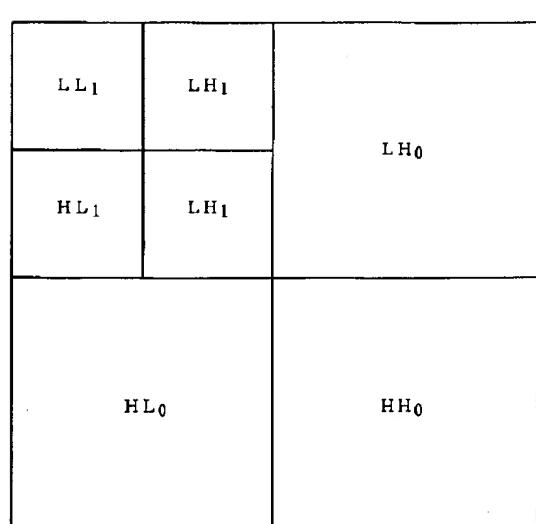
【図6】



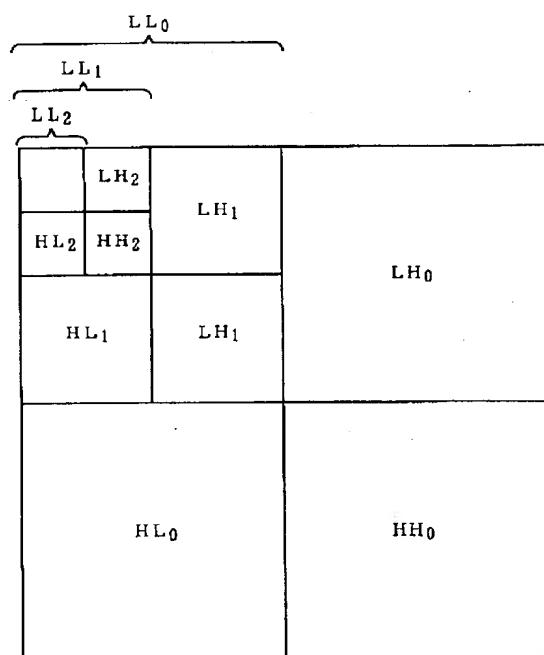
【図17】



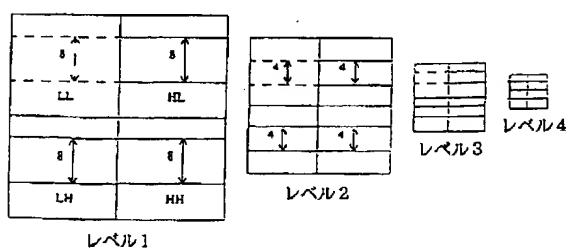
【図7】



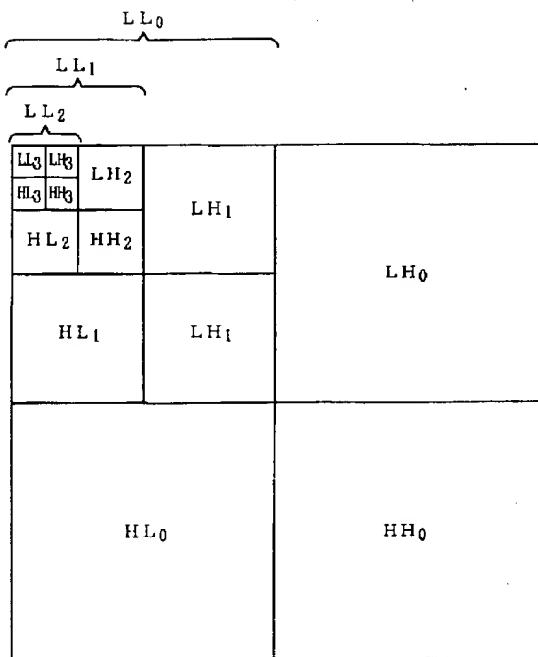
【図8】



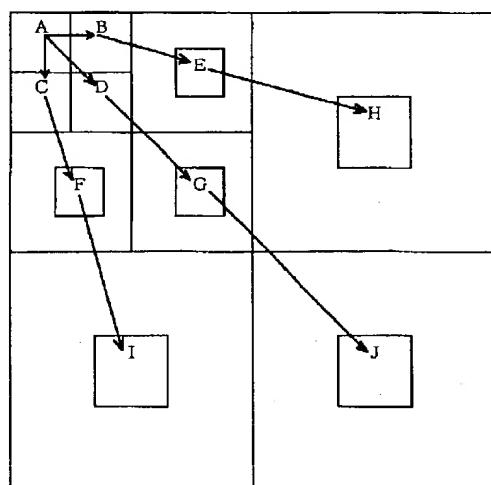
【図27】



【図9】

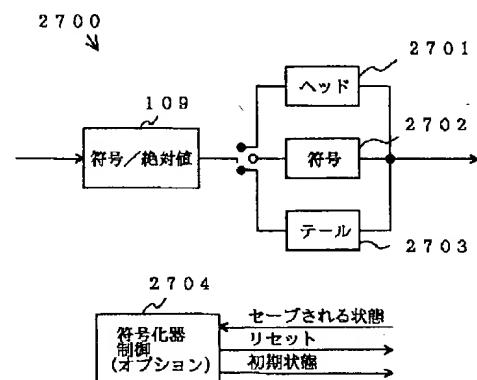
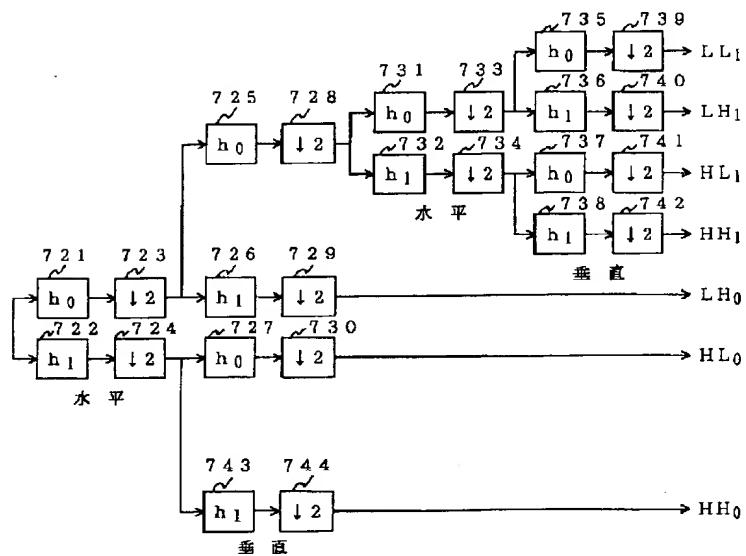


【図10】

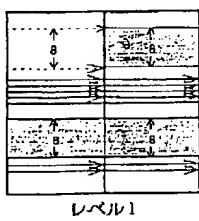
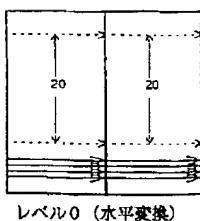


【図34】

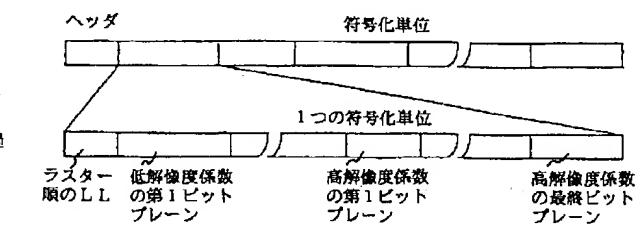
【図12】



【図29】

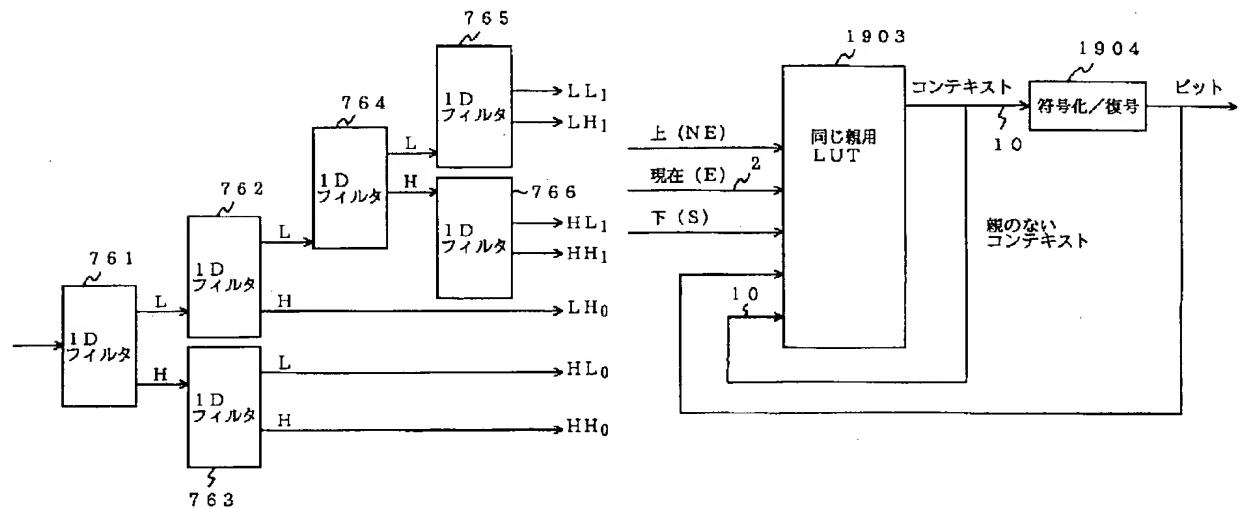


垂直
低域通過
垂直
高域通過

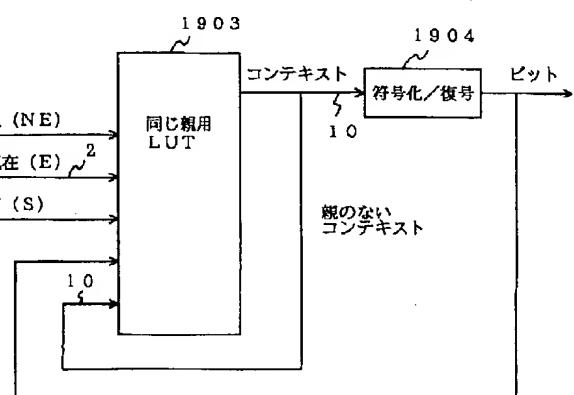


【図31】

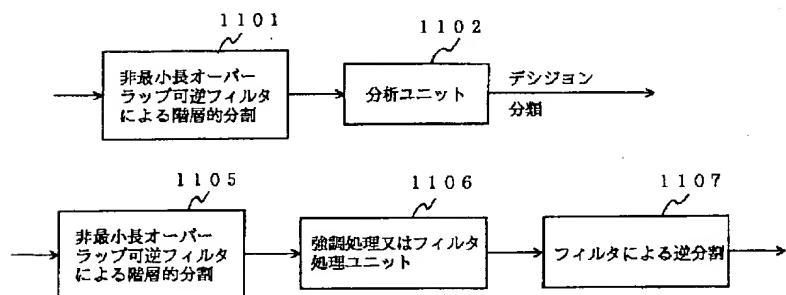
【図13】



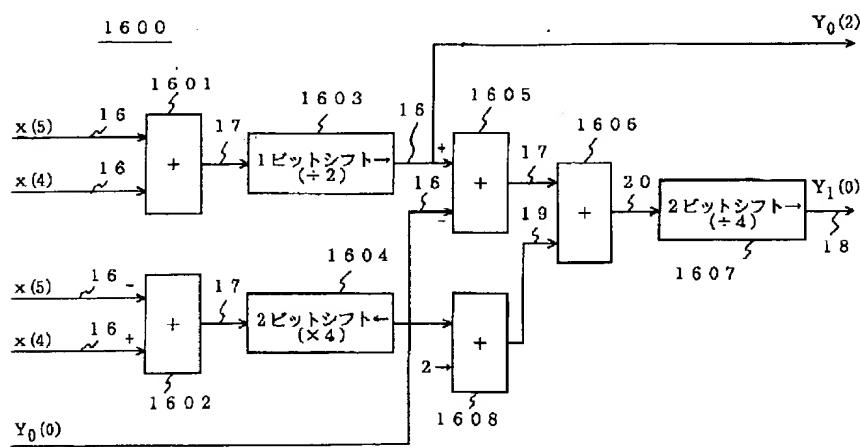
【図25】



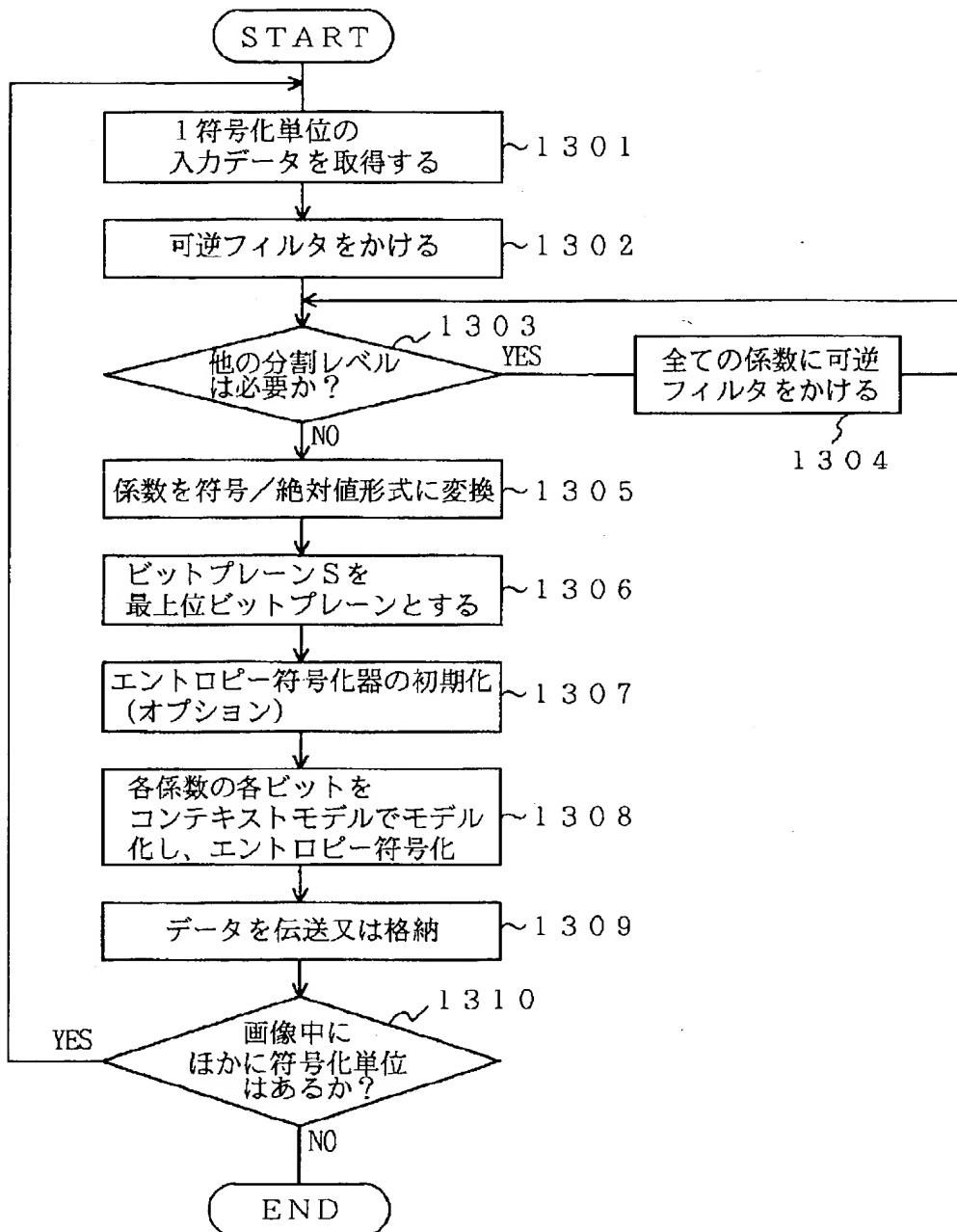
【図15】



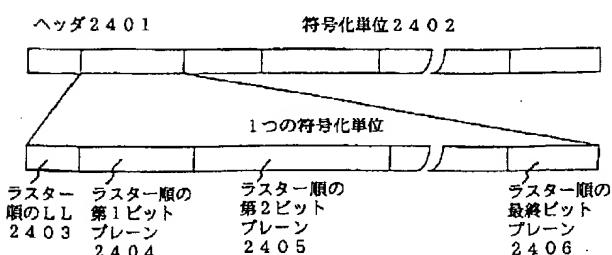
【図21】



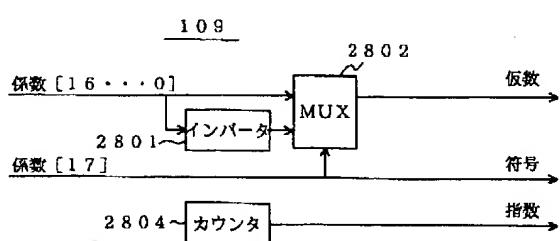
【図18】



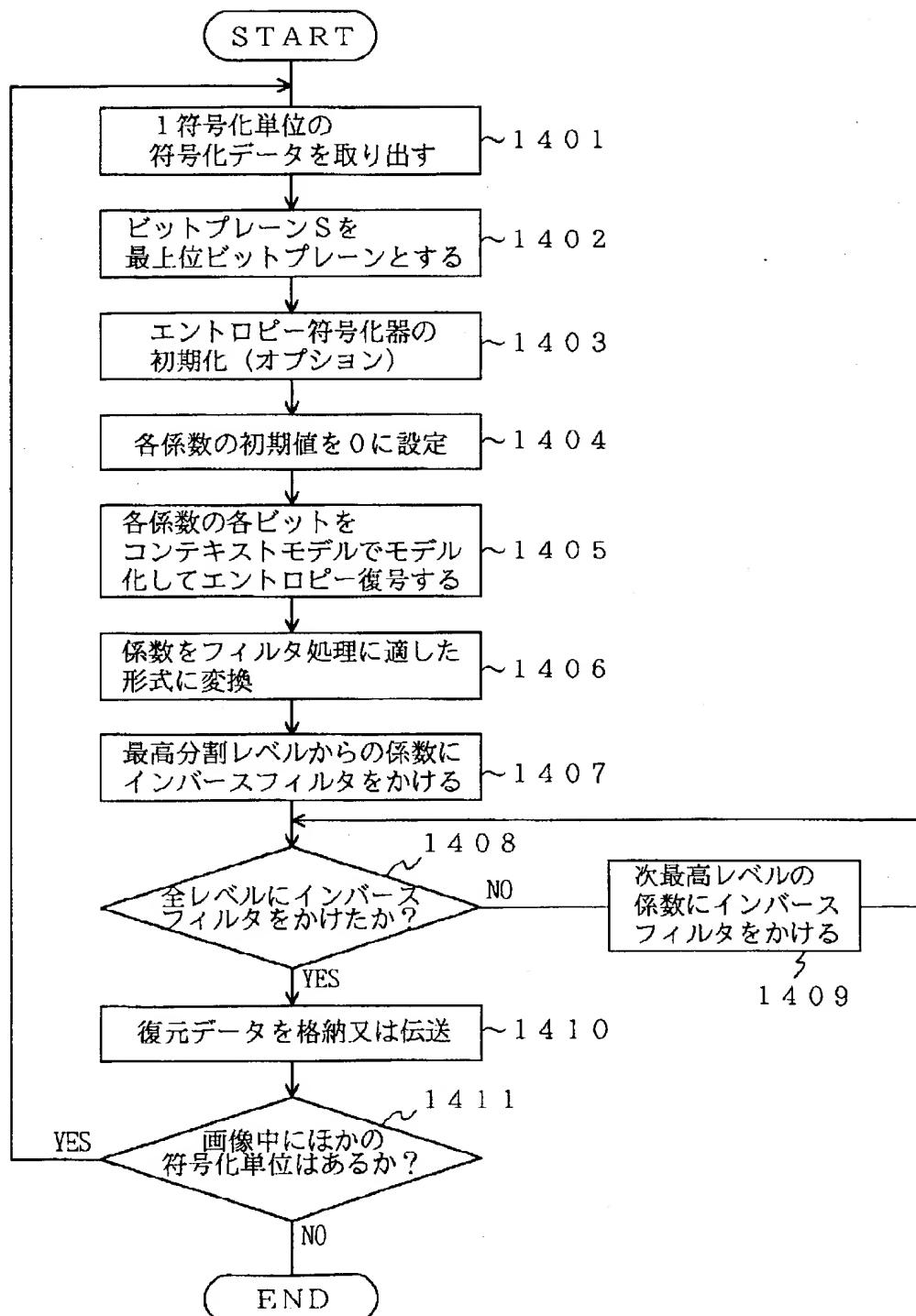
【図30】



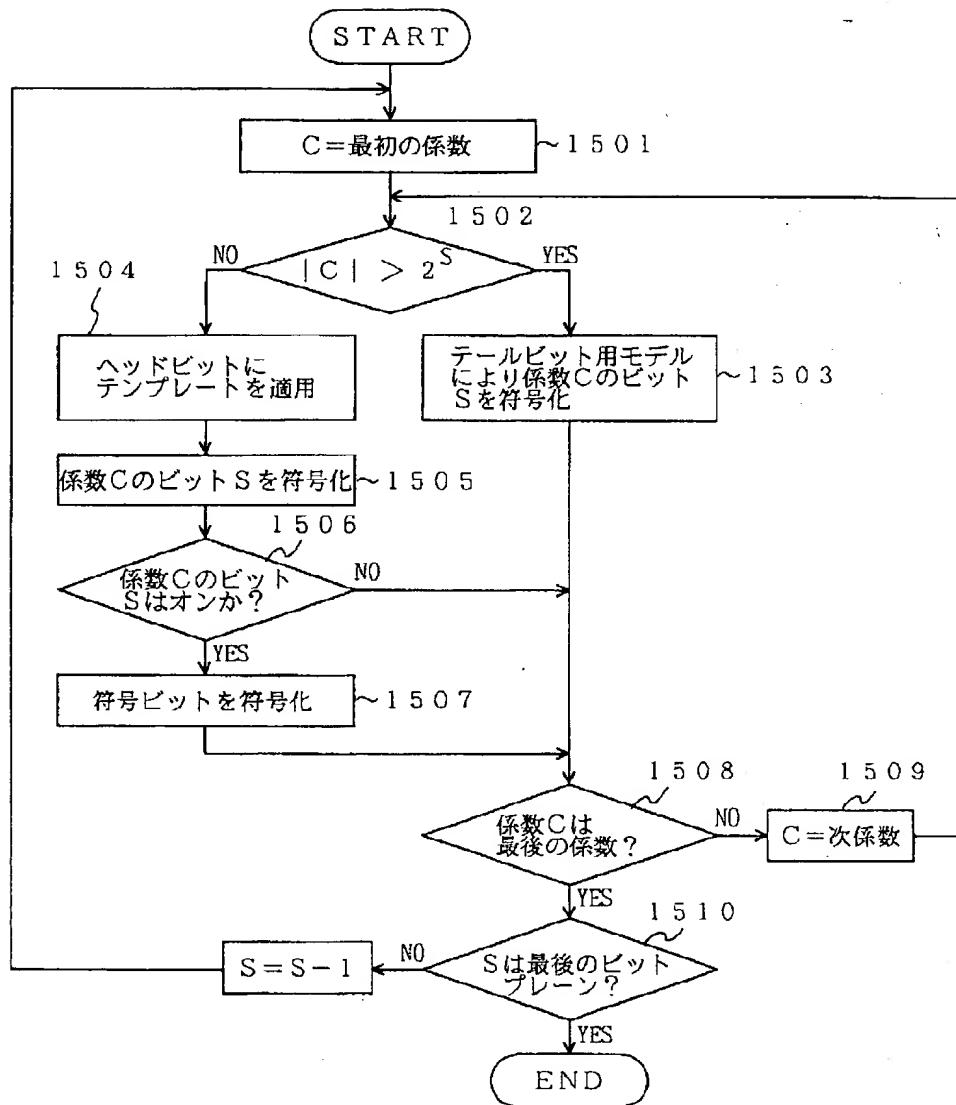
【図35】



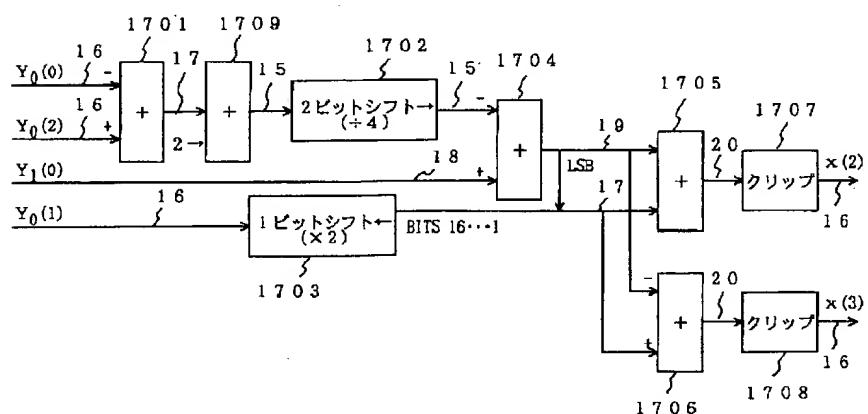
【図19】



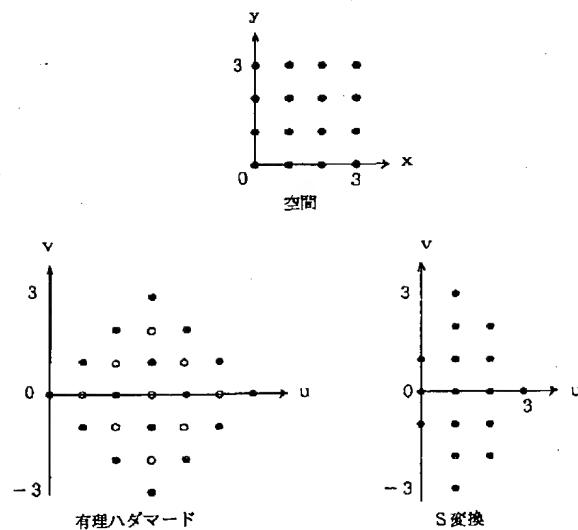
【図20】



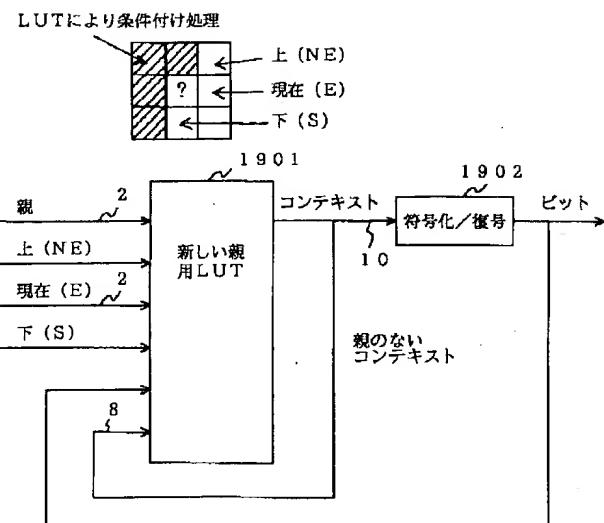
【図22】



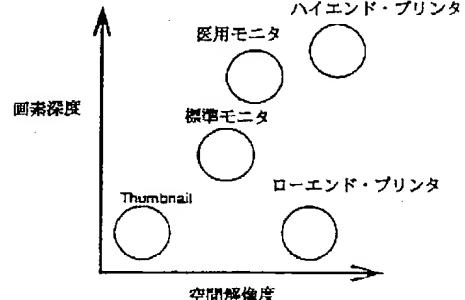
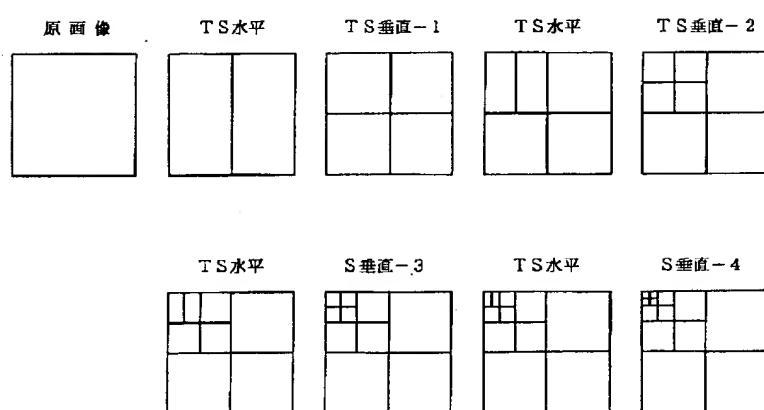
【図23】



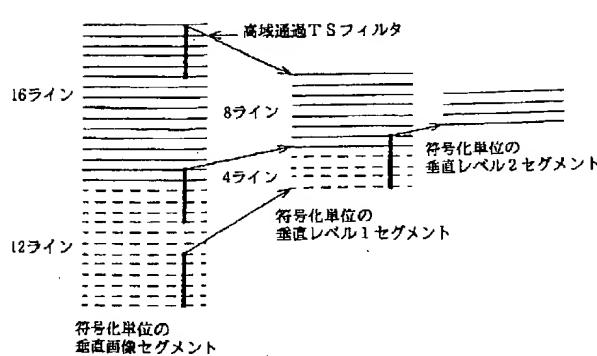
【図24】



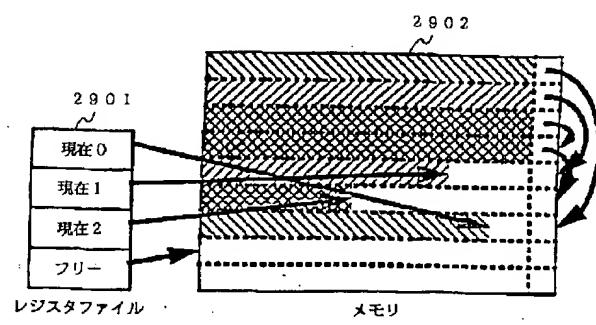
【図26】



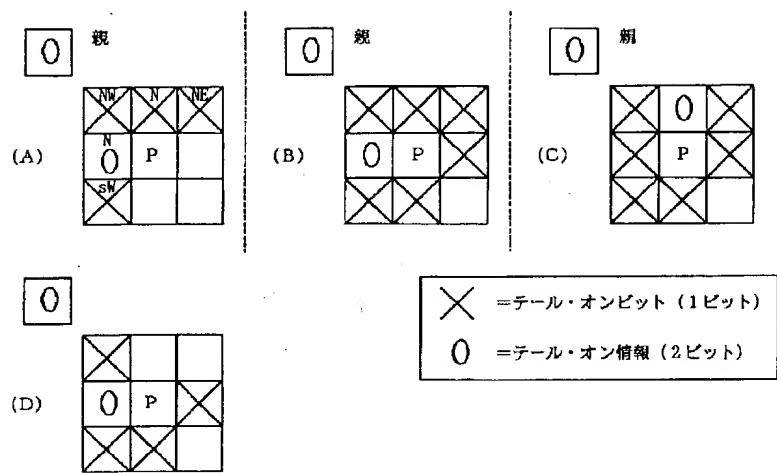
【図28】



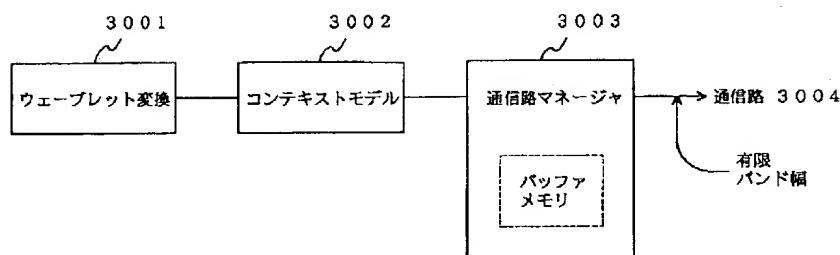
【図36】



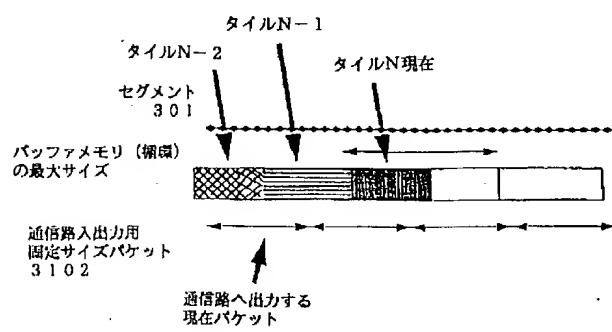
【図33】



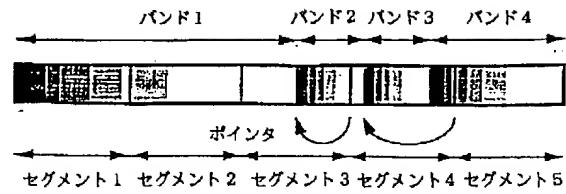
【図37】



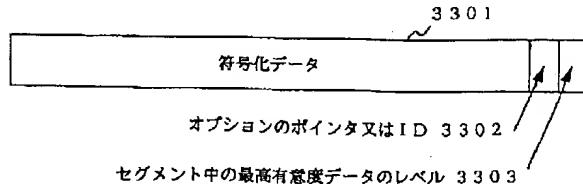
【図38】



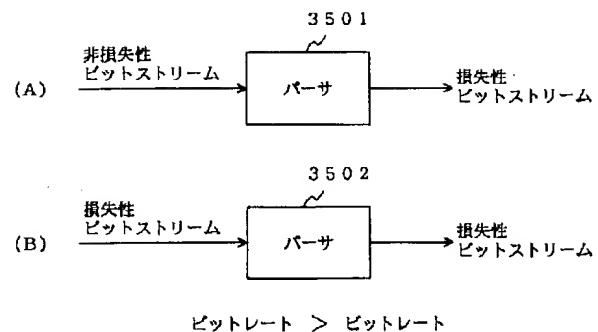
【図39】



【図40】



【図42】



フロントページの続き

(72)発明者 マイケル ジェー ゴーミッシュ
 アメリカ合衆国 カリフォルニア州
 94025 メンローパーク サンド ヒル 20
 ロード 2882 リコーコーポレーション内

(72)発明者 マーティン ボーリック
 アメリカ合衆国 カリフォルニア州
 94025 メンローパーク サンド ヒル
 ロード 2882 リコーコーポレーション内